



سلسلة هندسة المعلومات ونظم الكمبيوتر

الذكاء الاصطناعي

الاسحاب
للنشر والتوزيع

يحيى عبيد

دكتور: بشير على عرنوس

- الذكاء الاصطناعي - مقدمة
 - o أنظمة الذكاء الاصطناعي - تعاريف
 - o مثال العمل الذكي
 - o خلاصة
- أالذكاء الإصطناعي- مفاهيم رئيسية
 - o مشاكل وبحث
 - o "تعريف المشكلة
 - o "البحث
 - o "بحث إرشادي
 - o تمثيل المعرفة والبحث
 - o " رسوم بيانية (و-أو)
 - o " خوارزمية AO
 - o تحليل النهايات
 - o رضاء قيد
 - o لماذا هذه المواضيع مهمة؟
- لغة البرمجة Prolog و الذكاء الاصطناعي (كتاب DB & AI)
 - o برمجة الذكاء الإصطناعي
 - o "دعم للحساب الرمزي
 - o "دعم للبرمجة الإستطلاعية
 - o "اللغات الرئيسية للذكاء الاصطناعي
 - o أساسيات Prolog
 - o مصطلحات ومزايا Prolog
 - o "الهيكلية الأساسية للبيانات
 - o "أكثر حول مطابقة Prolog
 - o "الترّاجع
 - o "وجهات نظر تصريحية وإجرائية من البرامج
 - o التكرار
 - o "متابعة تنفيذ برنامج Prolog
 - o معالجة القائمة
 - o مواضيع باقية
- تمثيل المعرفة
 - o مادة التمثيل
 - o إستعمال المعرفة
 - o خصائص أنظمة تمثيل المعرفة
 - o مداخل إلى تمثيل المعرفة
 - o "معرفة علائقية بسيطة
 - o "معرفة موروثة
 - o معرفة إستنتاجية
 - o "معرفة إجرائية
 - o مسائل في تمثيل المعرفة
 - o الخلاصة
- التمثيل المنطقي للمعرفة
 - o المنطق المسند
 - o "مثال
 - o " علاقات الحالة
 - o تطبيقات وإمتدادات

- التمثيل الإجرائي للمعرفة
 - o التمثيل التصريحي و الإجرائي
 - o مثال
 - o كيفية استعمال المعرفة
- بنية الشق الضعيف والحشوة
 - o لماذا نستخدم تركيب البيانات هذا؟
 - o شبكات المعاني
 - o "التمثيل في شبكة المعاني"
 - o "الإستدلال في شبكة المعاني"
 - o "التوسع في شبكات المعاني"
 - o الأطر
 - o "التمثيل الهيكلي للمعرفة"
 - o "نفسى الاطر"
- بنية الشق القوي والحشوة
 - o التبعية التصورية
 - o المخطوطات
 - o الموسوعة CYC
- الأنظمة الخبيرة
 - o مقدمة
 - o نصمم النظام الخبيرة
 - o "المخطط المعماري للنظم الخبيرة"
 - o "اختيار المشكلة"
 - o "هندسة المعرفة"
 - o "أسئلة إمتحان"
 - o النظم الخبيرة والقواعد
 - o "مثال بسيط"
 - o "وسائل تفسير"
 - o "أنظمة أكثر تعقيداً"
 - o مثال نظام خبير في Prolog
 - o حالة للدراسة: MYCIN
- الاستنتاج مع التردد: استنباط مجهول
 - o ما هو الاستنتاج؟
 - o كيف نستنتج؟
 - o استنتاج متردد؟
 - o استنتاج غير متفق مع المقدمة
 - o "الاستنتاج الاصلي"
 - o "التحديد"
 - o "تطبيقات: أنظمة صيانة حقيقة"
- الاستنتاج الغير مؤكد: الطرق الإحصائية
 - o المقابل الومزي للاستنتاج الإحصائي
 - o طرق إحصائية أساسية -- إحتمال
 - o "نظرية Bayes"
 - o نماذج الاعتقاد وعوامل اليقين
 - o "الاستنتاج بعوامل اليقين"
 - o "التغلب على عيوب حكم Bayes"
 - o نماذج Dempster-Shafer

"حسابات Dempster-Shafer"

"جَمْعُ الإِعتقاداتِ"

شبكات Bayesian

"تطبيق"

"استنتاج في شبكات Bayesian"

" مثال عملي "

منطق ضبابي

"نظرية مجموعات ضبابية"

الاستنتاج المُوزَّع

مقدمة

أنظمة الاستنتاج المُوزَّع

أنظمة اللوح

"تنظيم المعرفة والاستنتاج"

"قضايا تطبيقية"

"حلّ المشكلة والاستنتاج"

"مثال الأنظمة"

أنظمة تمرير الوسائط

معالجة لغة طبيعية (كتاب DB & AI)

مقدمة

المشكلة

النحو

"كتابة القواعد"

"قواعد في Prolog"

"مُعرِّبات لغوية"

"مُراجعة شجرة الإعراب"

"اعراب مضاعف"

دراسة معاني الكلمات والفلسفة

"دراسة معاني الكلمات"

"فلسفة الذرائع"

التوالد

التخطيط

ماذا يتضمن التخطيط ؟

بحث في التخطيط

أمثلة موانع التخطيط

مكونات نظام التخطيط

"إختيار أفضل قانون"

"تطبيق نظام"

"إكتشاف التقدّم"

تخطيط مجموعة أهداف

"مبدأ Sussman"

تخطيط لاطفي باستخدام ايداع مقيد

"المساعدة على الكشف باستخدام ايداع مقيد"

التعلّم

ما التعلّم؟

"كَيْفَ نَتَعَلَّم؟"

روتين التعلّم

- "مخزن اختصارات الحساب
 - oالتَّعَلُّمُ بِأَخْذِ النَّصِيحَةِ
 - " مكننة أَخْذِ النَّصِيحَةِ
 - " صيانة قاعدة المعرفة
 - " مثال نظام التعلم - FOO
 - oالتَّعَلُّمُ بِحَلِّ الْمَشْكِلةِ
 - "التَّعَلُّمُ بِتَعْدِيلِ عَامِلِ الْمَقْيَاسِ
 - "التَّعَلُّمُ بِعَوَامِلٍ وَاسِعَةِ النِّطَاقِ
 - "التَّعَلُّمُ بِقِطْعٍ كَبِيرَةٍ
 - oالتَّعَلُّمُ الْاِسْتِقْرَائي
 - " مثال التعلم Winston
 - " حيز الوصف
 - " أشجار القرار
 - oالتَّعَلُّمُ الْمَبْنِي عَلَى التَّفْسِيرِ
 - " مثال
 - oاِكْتِشاف
 - "الاكتشاف بالنظرية المقادة AM
 - "الاكتشاف بالبيانات المقادة BACON
 - oالتناظر
 - "التناظر التحويلي
 - "التناظر الاشتقاقي
 - الادراك العام
 - oالعالم الطبيعي -- فيزياء نوعية
 - "نمذجة العالم النوعي
 - " الاستنتاج مع المعلومات النوعية
 - oالادراك العام لعلم الوجود
 - "الزمان
 - "المكان
 - "المادة
 - oتنظيم الذاكرة
 - oالذاكرة في حل المشكلة
 - الرؤية
 - oتحدي الرؤية
 - oاِكْتِساب الصورة
 - " ادخال صورة ثنائية الابعاد
 - " الصور ثلاثية الابعاد
 - oعرض هندسي لرؤية الحاسوب
 - "نماذج الاطر السلوكية
 - "نموذج السلسلة التامة
 - "تمثيل الحد الفاصل
 - "مواصفات نموذجية مرغوبة للرؤية
 - oالتصنيف الخطي
 - "مقدمة
 - "فرضيات
 - "أنواع نقاط الالتقاء
 - "تصنيف الصورة

oالتصنيف الإرخائي
"التقنية الإحصائية"
oتمييز الأشياء
"الثوابت"
"تمييز النقوش"
"تحويّلات Hough"
"صور Gaussian"
oنموذج ميني على اساس تمييز الأشياء
"طرق شجرة البحث"
"بعض دراسات السيرة المماثلة"
"طرق التصنيف الارخائي"
"بحث الرسم البياني"

بسم الله الرحمن الرحيم

الحمد لله رب العالمين الذي علم الإنسان مالم يعلم، والصلاة والسلام على رسوله الكريم أكمل خلقه. خلق الله الإنسان بإحسن تقويم، وميزه عن كل مخلوقاته بالعقل وحرية إتخاذ القرار (وعرضنا الأمانة... وحملها الإنسان). وكرمه بأن جعله خليفته على الارض، واستوجب هذا الاستخلاف، الإبداع عند الانسان. عمل الإنسان وابدع في عمارة الأرض مستندا على المعرفة التي تعلمها من خالقه (وعلم آدم الاسماء)، كذلك المكتسبة من تجارب جنسه المتركمة. خلال القرون الطويلة لخلافة الإنسان على الأرض، وباستخدام ميزة العقل لديه، استطاع ولايزال يدرس نفسه وكل ماحوله من المخلوقات. أسس الإنسان قاعدة معرفة ضخمة في مختلف العلوم كالرياضيات والفيزياء والطب والفلك... الخ. في كل إبداعه كان الانسان ولايزال يحاول ان يقلد خالقه، وشتان ما بين إبداعات الخالق والمخلوق. ولأن إبداعات الخالق تتصف بالكمال، فما كان من الانسان الا ان يحاول فهم هذه المخلوقات، ومن بينها هو نفسه. وان يستخدم هذه المعرفة في انتاجاته وإبداعته.

يتجلى هذا التقليد، بشكل واضح، في محاولات الإنسان تطوير الحاسبات الآلية وعلى رأسها انظمة الذكاء الاصطناعي. حيث يقف الإنسان على رأس هرم المخلوقات، يتميزه عنها جميعا بالعقل والذكاء. بالرغم من أن نسبة الذكاء لدى افضل نظام إصطناعي، لاتصل الى ذكاء طفل بعمر الـ 12 سنة، لكنها مفيدة للإنسان. قام الإنسان بتطوير هذه النظم بالإستناد على تكوين جسده وعقله. تقوم هذه النظم بتنفيذ الاعمال الرتيبة والمجهددة للإنسان، تستبدله احيانا في الأماكن الخطرة صحيا، والاماكن الصعبة البلوغ بالنسبة اليه. تكون مساعدا وشريكا له في تحليل و/ أو حل المشاكل. بناءا على ذلك عمل الانسان ولازال يعمل على كشف طريقته في التعلم والادراك والتفكير والاستنتاج واتخاذ القرار. بالرغم من انه لم يصل بالتحديد الى الطريقة التي يقوم هو نفسه بتمييز الأشياء، لكنه استطاع ان يقدم العديد من الفرضيات والمبادئ والطرق لتنفيذ بعض انواع التفكير والاستنتاج. كذلك قام ببناء العديد من أنظمة الذكاء الاصطناعي التي تهتم بحل مشاكل محددة.

تعتبر مداخل المعلومات لدى الإنسان- الحواس الخمس : البصر، السمع، الشم، الذوق و الحس. ويعتبر العقل هو المعالج الرئيسي المدير لعمل باقي الاجزاء والمحلل للمعلومات المدخلة والمسؤول عن التفكير والإدراك والإستنتاج و إتخاذ القرار. تعتبر الذاكرة (أجزاء من المخ) مركز تخزين المعرفة بكل انواعها مع خاصية استرجاعها عند الحاجة. يعتبر نصيب حاسة البصر لدى الإنسان، من نسبة المعلومات المدخلة عن طريقه، الاكبر، حيث يبلغ 90% من إجمالي المدخلات. بينما تتوزع الـ 10% الباقية على الحواس الأربعة الأخرى. لذلك نجد ان الإنسان ركز على موضوع البصر الصناعي، لجعله المدخل الاساسي لمعلومات نظمه المطورة. تتكون هذه الانظمة من نفس الوحدات الرئيسية نفسها التي يملكها الإنسان. فنجد فيها وحدات الإدخال : كآلات التصوير، الماسح الضوئي، لوحة المفاتيح، أجهزة التقاط الصوت، الفأرة... الخ. كما نجد وحدة المعالجة المركزية (المعالج)، ووحدات الذاكرة بمختلف انواعها. بالإضافة الى كل هذه الاجزاء الصلبة نجد البرمجيات التي تعتبر مسؤولة عن إدارة الاجزاء الصلبة وتحليل المعلومات المدخلة، تخزينها، ومعالجتها للوصول الى مخرجات الحل.

هذه البرمجيات مبنية على مختلف الاسس والمبادئ والنظريات الخاصة بإدخال وتشكيل قواعد البيانات وحفظ المعلومات، تمثيل المعلومات، المعالجة المبدئية للمعلومات بهدف تحسينها، معالجة المعلومات لإستخراج دلائل التمييز بالإعتماد على مبدأ التمييز المستخدم، ترجمة المعلومات للغة الدلائل، معالجة المعلومات لحساب معايير المطابقة بعد إختيارها حسب المسألة المطلوبة، مطابقة الدلائل وإعطاء القرار (النتيجة). يغطي هذا الكتاب بنسب متفاوتة كل مراحل وأدوات وطرق تطوير انظمة الذكاء الإصطناعي، حيث ان بعض هذه المراحل، سيخصص لها كتب منفصلة لشرحها بالتفصيل. يبدأ الكتاب بمقدمة تعرف القارئ بتعاريف هذه النظم وأنواعها ومهامها. ينتقل الى طرق تمثيل المعرفة ومشاكل البحث. يعطي فكرة عن لغات البرمجة المستخدمة في نظم الذكاء الإصطناعي ومعالجة اللغة الطبيعية. ينتقل الى الشرح المفصل عن الاستنتاج وأنواعه وطرقه. يوضح بتفصيل التخطيط واساليبه لتطوير نظم الذكاء الإصطناعي. يشرح بتمعن التعلم بمراحله وطرقه وخصائصه لهذا النوع من الانظمة. يعرف بالتفكير والادراك العام والاستنتاج المميز لهذه الانظمة. وأخيرا يقدم المعلومات الكافية عن خاصية البصر المعطاة لهذه الانظمة لرؤية وتمييز الاجسام والاشياء، مع إعطاء الشرح والتفاصيل وبعض الامثلة عن نظم ومبادئ التمييز. مع العلم بأنه سيخصص كتاب مفصل لمعالجة وتمييز الصور (سيغطي بالتفصيل الرؤية ونظم التمييز).

تم إعتداد اسلوب الملخصات في هذا الكتاب، ليكون اقرب الى الكتاب الأكاديمي، ويكون اسهل للقراءة والفهم من اسلوب السرد. تم طرح بعض الامثلة من الحياة العادية، لتساعد على فهم المضمون لبعض الطرق والمبادئ. هذا الكتاب موجه للدارسين والباحثين باللغة العربية والعاملين والمطورين والمبرمجين في مجالات تقنية المعلومات من هندسة الكمبيوتر ونظم المعلومات والنظم الطبية، المجالات الهندسية الأخرى، المجالات الطبية، المجالات العلمية الأخرى كالرياضيات، الفيزياء، الكيمياء، الجغرافيا، علوم الارض، علوم الفضاء... الخ. كذلك العلوم الإجتماعية بأنواعها.

لمحة عن المؤلف

د. بشير علي عرنوس من مواليد دمشق 1966، دكتوراة في هندسة الحاسب الآلي، تخصص انظمة معالجة وتمييز الصور. عمل في مجال التعليم الأكاديمي في جامعة فلاديمير الحكومية التقنية، فلاديمير – روسيا. كما عمل في مجال تطوير نظم المعالجة والتمييز (نظام قارئ آلي للغة العربية). واخيرا يعمل في ادارة مشاريع تقنية المعلومات. يسر المؤلف معرفة آراء القراء و استلام كل الملاحظات، الاقتراحات والاستفسارات حول الكتاب، على البريد الإلكتروني : drarnous@hotmail.com .

المقدمة

لمحة عن تاريخ الذكاء الاصطناعي

فكرة تشكيل الذكاء الاصطناعي، على نحو مشابه للخاصية البشرية، لحل المسائل الصعبة و نمذجة الذكاء الطبيعي (البشري)، خرجت للضوء في العصور القديمة. فمثلا عند المصريين القدماء تم اختراع تمثال ميكانيكي لآمون (له خاصية تحريك بعض اعضائه). في الإليادا لجومير تم وصف "جوفيست" كآلة على شكل الإنسان. تم طرح هذه الفكرة بشكل كبير في المنتجات الادبية ابتداء من: دمية بيغمليون وحتى اللعبة الخشبية لبوب كارلو. يعتبر الفيلسوف الاسباني رايمود لولي من القرون الوسطى (القرن الثالث عشر) هو المؤسس الاول لفكرة الذكاء الاصطناعي، حيث انه حاول تصميم آلة ميكانيكية لحل مشاكل متعددة، على اساس مفهوم التصنيف العام الذي وضعه بنفسه. في القرن الثامن عشر اقترح كل من ديكارت و ليبنتس فكرة لغة عامة لتصنيف كل العلوم. هذا العمل يمكن اعتباره اول عمل نظري في مجال الذكاء الاصطناعي. تمت ولادة فكرة الذكاء الاصطناعي كإتجاه علمي محدد بالنهاية بعد تصميم اجهزة الكمبيوتر في اربعينيات القرن الماضي.

تم اقتراح مصطلح الذكاء الاصطناعي (artificial intelligence) في عام 1956 في احدى الندوات العلمية في احدى الجامعات الامريكية. دارت هذه الندوة حول تطوير مبادئ لحل مشاكل منطقية وليس حسابية آنذاك. في اللغة الانجليزية تعني كلمة intelligence إمكانية التفكير بذكاء.

بعد الإعراف بالذكاء الاصطناعي كمجال علمي منفصل، تم تقسيمه الى اتجاهين :

- شبكات البحث العصبية
- شبكات البحث – الصندوق الأسود

ولادة شبكات البحث العصبية

يمكن تشكيل الفكرة الاساسية من هذا الاتجاه كالتالي:

- جسم وحيد له خاصية التفكير والاستنتاج – العقل البشري.
 - لذلك اي جهاز سيملك خاصية التفكير يجب ان يقلد نفس بنية العقل البشري.
- لذلك كانت مهمة شبكات البحث العصبية موجهة لتطوير بنى (انظمة – برمجيات + اجهزة) تشبه بنية العقل البشري. يتكون العقل البشري من مجموعة كبيرة من الخلايا العصبية (10²¹) المرتبطة والمتفاعلة مع بعضها البعض. على هذا النحو اهتمت شبكات البحث العصبية لتكوين عناصر مماثلة للعصبونات، وربطهم ببعض ضمن النظام الوظيفي. تسمى هذه النظم بالشبكات العصبية. اول شبكة عصبية صممت من قبل روزن بلاتوم و ماك كالاكوم بين العامين 1956- 1965. وكانت محاولة لتقليد البصر البشري و تفاعله مع المخ البشري. تمكن هذا النظام من تفريق

احرف الهجاء، غير انه كان شديد الحساسية لتغير الخط، بحيث انه اعتبر مثلا احرف A المكتوبة بخطوط مختلفة اجسام مختلفة.

بعد ذلك تقلص عدد المحاولات في هذا المجال، الى ان تمكن الخبراء اليابانيون في الثمانينات من تصميم اول كمبيوتر - عصبي ضمن مشروع كمبيوتر الجيل الرابع. حيث لهذا الوقت تم التغلب على مشكلتي سرعة تنفيذ العمليات وحجم الذاكرة في اجهزة الكمبيوتر. وتم اختراع الترانسيبوتر بخاصية الحل المتوازي باستخدام مجموعات من المعالجات التي تعمل بشكل متوازي.

تعتبر المهمة الاساسية لشبكات البحث العصبية هذه الايام تمييز الاجسام من الصور المأخوذة من الفضاء. هناك ثلاث طرق لتصميم شبكات البحث العصبي:

- مستوى الاجهزة - بتصميم اجهزة كمبيوتر بمعالجات إضافية لتنفيذ كافة خوارزميات المعالجة والتمييز.
- مستوى البرمجيات - تطوير برمجيات تقوم بتنفيذ كافة مهام الخوارزميات وتنفذ على حواسيب قوية.
- مستوى المزيج - بحيث نمزج الطريقتين السابقتين فبعض المهام تنفذ باستخدام معالجات مساعدة والاخرى باستخدام برمجيات.

من شبكات البحث حتى الذكاء الاصطناعي

ترتكز هذه الطريقة على اتجاه معاكس للطريقة الاولى (الشبكات العصبية)، بحيث لايهم هنا كيفية بناء النظم المفكرة. المهم هنا ان يكون رد الفعل على البيانات المدخلة مشابه لرد فعل المخ البشري. مشجعي هذا الاتجاه ارتكزوا على ان الانسان لايجب ان يتبع وبشكل اعمى التصميمات الطبيعية، واعطوا مثالا على ذلك : نجاح "الدولاب" والذي لم يتوفر بالطبيعة بشكله المعروف ووظيفته. بالإضافة الى الجهل الكبير بطرق (العمليات الداخلية ضمن العقل) تفكير الإنسان واستنتاجاته وكيفية بناء الذاكرة.

هذا الاتجاه من الذكاء الاصطناعي اهتم بالبحث عن الخوارزميات الخاصة بحل مشاكل الذكاء الاصطناعي باستخدام نماذج الحواسيب المتوفرة. ان المساهمة الفعالة في هذا المجال كانت تطوير لغة برمجة خاصة بأنظمة الذكاء الاصطناعي (LISP) من قبل ماكارتي. ايضا اكتشاف الاطر و نموذج الاطر لتمثيل المعرفة. جرى بحث مكثف ما بين الاعوام 1956 - 1963 لنماذج وخوارزميات التفكير الإنساني وتطوير اول البرمجيات المبنية على اساس هذه الخوارزميات. تمكن خبراء الذكاء الاصطناعي من عمل هذا عندما لم يستطيع وقتها علماء العلوم الاجتماعية (الفلاسفة، علماء النفس، علماء اللغة) تقديم هذه الخوارزميات. عليه وبشكل متسلسل تم تطوير مختلف الطرق التالية:

- صمم نموذج بحث لابرينت بنهاية الخمسينات من القرن الماضي، يتمثل هذا المبدأ بإيجاد الطريق الأقصر للحل الامثل خلال مخطط البحث (تمثيل المخطط). لم يحظى هذا المبدأ باستعمال واسع. طبقت على اساسه بعض الألعاب كالشطرنج، وبناء برج هانوي.
- صمم نموذج على اساس اختصار مجموعة الخيارات في البحث (بداية الستينات).
- تصميم مبدأ العزل باستخدام رياضيات المنطق، 1963-1970. بناء على هذا المبدأ تم تطوير لغة البرمجة Prolog.
- تطوير عدة انظمة خبيرة مثل MYCIN و DENDRAL 1976-1978 بالإعتماد على تضيق مجال المعرفة والتخصص في مجال واحد مثلا الطب او الكيمياء كما في الامثلة المعطاة.
- تطوير نظم وحواسيب الجيل الرابع في الثمانينات من القرن الماضي.
- من منتصف الثمانينات وحتى الان تضاعف بشكل كبير عدد الانظمة الخبيرة والنظم الذكية وغطت الكثير من مجالات الحياة ومختلف العلوم. كأمثلة على ذلك يمكن عرض التالي:
- أنظمة القراءة الآلية لحروف اللغات المختلفة.
- أنظمة مضاهاة التواقيع
- أنظمة مطابقة البصمات
- أنظمة مطابقة بصمة العين
- النظم الخبيرة الطبية
- النظم الخبيرة الخاصة بالصيانة الخ.

أنظمة وتعريف الذكاء الاصطناعي

التعريف العام والجيد للذكاء الاصطناعي يُمكن أن يكون:
الذكاء الاصطناعي جزء من علم الحاسبات والذي يُهتَم بأنظمة الحاسوب الذكية، تلك، الأنظمة التي تمتلك الخصائص المرتبطة بالذكاء واتخاذ القرار والمثابرة لدرجة ما للسلوك البشري في هذا المجال – فيما يخص اللغات، التعلّم، التفكير وحلّ المشاكل... الخ.

انظمة الذكاء الاصطناعي : حقل واسع، وهي مُهمّة بتطوير الحاسبات لتقوم بتنفيذ المهام التي تتطلّب ذكاءاً إنسانياً. هناك العديد من هذه المهام، مثل الحساب المعقّد. الحاسبات الآلية يُمكن أن تُنفذ هذه المهام بسهولة شديدة. بالمقابل، هناك العديد من المهام التي يقوم بها الناس حتى بدون تفكير - مثل تمييز الوجوه - والذي يعتبر معقداً جداً للأتمتة (المكننة). انظمة الذكاء الاصطناعي مُهمّة بهذه المهام الصعبة، التي تتطلّب العمليات المعقدة والمعرفة المعقدة والمتطورة.

يقوم الناس بتطوير انظمة الذكاء الاصطناعي و أتمتة المهارات الإنسانية لعدد من الأسباب المختلفة. أحد الأسباب ببساطة هو الفهم الاوسع للطريقة الإنسانية في التفكير واتخاذ القرار. على سبيل المثال، نحن قد نكون قادرين على إختيار وتقييم النظريات النفسية واللغوية بكتابة البرامج التي تُحاوّل تقليد مظاهر السلوك البشري. السبب الآخر ببساطة نحن نهتم ان يكون عندنا برامج ذكية، دون التقيد بمجاراتها لطرق التفكير الإنساني بدقة، مفيدة لحلّ المشاكل الصعبة.

هناك من يعتقد بأن الذكاء الاصطناعي يمكن وصفه بالحقول الذي يتداخل بعلم الحاسبات بدلاً من أن يكون حقل ثانوي منها، حيث انه أكثر وثوقاً للصلة إلى علم النفس، الفلسفة، المنطق، علوم اللغة، وحتى علم وظائف الاعضاء. على أية حال، يعتبر الذكاء الاصطناعي جزء من علوم الحاسبات حيث انه يستخدم تقنيات وادوات الحسابات، لتنفيذ عمليات مكننة الخوارزميات المبنية على اساس العلوم المذكورة اعلاه. على سبيل المثال يمكن سرد بعض من القضايا الفلسفية: بعض مهام الذكاء الاصطناعي

يمكن تقسيم الذكاء الإنساني الى نوعين من الوظائف : ``حياتية، ذكية`` و ``خبيرة``. بالحياتية الذكية أعني كل تلك المهام التي (تقريباً) كلنا يمكن أن نقوم بها بشكل دوري (بدرجات مختلفة) لكي نتصرف ونفاعل في العالم. هذا يتضمن:

- الوؤية: الرؤية مع القدرة على فهم الذي نراه.
 - اللغة الطبيعية: القدرة على الإتصال مع الآخرين في اللغة الطبيعية العربية، الإنجليزية أو غيرهم.
 - التخطيط: القدرة على تخطيط سلسلة من الأعمال لنيل الأهداف المرجوة.
 - الحركة: القدرة على التحرك والتصرف بالحياة، لتنفيذ المتطلبات الحياتية.
- تقوم النظم الذكية على عدد كبير من الحسابات المعقدة، التي تعالج فيها الصور المدخلة، والتي على الأغلب، تكون صور جرافيك. ومن ثمة يتم إختيار مبدأ التمييز وعلى اساسه تختار دلائل التمييز ويتم حسابها ومطابقتها مع تلك التي مخزنة في قاعدة البيانات. بالإعتماد على نتيجة المطابقة، يتم اتخاذ القرار، بإنتماء الجسم المطابق الى واحد من الاصناف المحتملة.

بالخبيرة نعني المهام التي فقط بعض الناس جيّدون في تنفيذها، والتي تتطلّب تدريب شامل. والتي يمكن أن تكون مفيدة خصوصاً لأتمتة هذه المهام، حيث يمكن ان يكون هناك نقص بالخبراء الإنسانيين. كمثال للتفكير الخبير :

- التشخيص الطبي.
 - تصليح الأجهزة.
 - ترتيب الحاسوب.
 - التخطيط المالي.
- الأنظمة الخبيرة مُهمّة بأتمتة هذه الأنواع من المهام. والتي على الأغلب تكون مدخلتها نصية، مع إمكانية احتوائها على صور بهدف التوضيح والاستدلال. إتخاذ القرار في هذا النوع من النظم، يتم بناءاً على الخبرات المدخلة في قواعد بيانات هذه النظم من قبل الانسان الخبير، وليس بناءاً على حسابات كما في النوع الاول.
- الذكاء الاصطناعي مُهمّت بأتمتة النوعين. مع العلم، بأن المهام الحياتية الذكية هي الاصعب للأتمتة.

هناك العديد من التعاريف والتقسيمات الأخرى حسب مراحل انظمة الذكاء الاصطناعي، حيث يمكن تقسيم هذه المهام الى :

- البحث.
- تمثيل المعرفة.
- التطبيقات على البحث والتمثيل.

تتطلب التقنية العامة الحاسمة في كتابة برامج الذكاء الاصطناعي الى بحث. في أغلب الأحيان ليس هناك طريق مباشر لإيجاد حلولٍ لهُغْض المشاكل ، حتى وان كنا قادرين على وضع الاحتمالات. على سبيل المثال، في حل مشكلة ما قد نعرف كل الحركات المحتملة، لكن ليس التسلسل التي تؤدي إلى الحل. لنفترض انك تريد الذهاب لمكان ما، و أنت تعرف كل الطرق الممكنة لذلك سيارات / حافلات / قطارات... الخ، لكنك لاتعرف أفضل طريق للوصول إلى الجهة المطلوبة بسرعة. لذا يتوجب البحث ضمن هذه الإمكانيات لإيجاد انجع الحلول. هناك العديد من التقنيات الخاصة بالبحث والتي سيتم طرقها بالتفصيل لاحقاً.

يتعامل تمثيل المعرفة مع إيجاد وسائل تفسير المعلومات الخاصة بالمعرفة لكي يمكن أن تستعملها الحاسبات لاحقاً. سيتم الحديث بالتفصيل عن تمثيل المعرفة في الفصل الخاص بهذا الموضوع.

كما سيبحث هذا الكتاب مواضيع مهمة من الذكاء الاصطناعي مثل التخطيط، التفكيك والاستنتاج، التعلم، الفهم و التمييز. كذلك شيناقتش مهام مثل الرؤية، فهم اللغات الطبيعية، تمييز الصوت والتخطيط الآلي.

الخلاصة

يمكن استنتاج الخلاصة التالية من هذه المقدمة : كل مهام حل المشاكل المنوطة بانظمة الذكاء الاصطناعي يمكن أن تتلخص بنوعين:

تمثيل المعرفة، و البحث

تعريف المشكلة

لتوضيح المشكلة سنأخذ مثال: لعبة بين شخصين على طاولة لعب على سبيل المثال لعبة الشطرنج ، مبدائياً علينا تحديد قواعد اللعبة والأهداف المفروض الوصول لها للفوز، بالإضافة إلى وسائل تمثيل المراحل في اللعبة. المرحلة الافتتاحية يمكن أن تعرف كالحالة الأولية (صف الاحجار على رقعة اللعب)، و مرحلة الفوز كحالة الهدف (سقوط احد الملكين)، يمكن أن يكون هناك أكثر من هدف واحد. تسمح الحركات في اللعبة بالانتقال من الحالة الأولية إلى الحالات الأخرى بما يؤدي في النهاية الى الوصول إلى حالة الهدف. على أية حال وبالرغم من ان القواعد في لعبة كالشطرنج محدودة العدد، غير ان الطرق والخطط للعب غير منتهية العدد. هذه القواعد لا تستطيع أن تكون مجهزة عموماً بدقة، وبرامج الحاسوب لا تستطيع معالجتهم بسهولة. كما تواجهنا مشكلة أخرى متعلقة بوسط التخزين، غير ان البحث يمكن أن ينجز بالجزئية.

إن عدد القواعد المستعملة يجب أن يقلل للحد الأدنى حسب المتاح، والمجموعة يمكن أن تعامل بمعالجة كل قاعدة في المجموعة على حدى بشكل عام. إن تمثيل الألعاب بهذه الطريقة يؤدي إلى تمثيل فضاء رسمي وهو طبيعي للألعاب المنظمة بشكل جيد ببعض التراكيب. هذا التمثيل يسمح لنا بالتعريف بالمشكلة والذي يستوجب الحركة من مجموعة المواقع الأولية إلى إحدى مجموعات مواقع الهدف. يعني بأن الحل يتضمن استعمال تقنيات معروفة والبحث المنظم.

- مشاكل منظمة بشكل جيد (ومثال على ذلك: - ألعاب) يمكن أن توصف كمجموعة القواعد.

- قواعد يمكن أن نعلم ونمثل كتمثيل فضاء رسمي:

o تعريف رسمي.

o التنقل من المرحلة الأولية إلى إحدى مجموعات مواقع الهدف.

o تحرك منجز عن طريق بحث منظم.

البحث

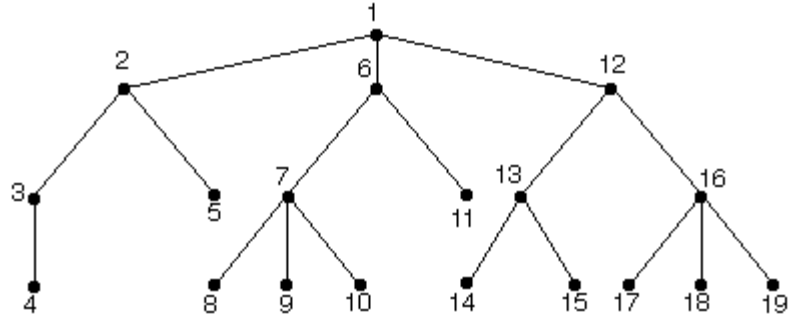
هناك طريقتان أساسيان لإداء البحث:

- بحث اعمى -- يمكن فقط أن يتحرك طبقاً للموقع في البحث.
- بحث إرشادي - باستخدام معلومات معينة لتقرير منطقة البحث.

البحث الأعمى - العمق اولا

1. نعتبر المجموعة L قائمة العقد الأولية في المشكلة.
2. إذا كانت L فارغة، نصل الى فشل من العقد الأولى، في خلاف ذلك نأخذ العقدة n من L.
3. إذا كانت العقدة n هي حالة الهدف، نعاود الكرة من العقد الأولية.

4. ما عدا ذلك تحذف الخطوة n من L ويتم اضافة كل اطفال ns إلى بداية L . ضع علامة لكل طفل مع طريقه من العقدة الأولية. عد إلى 2.

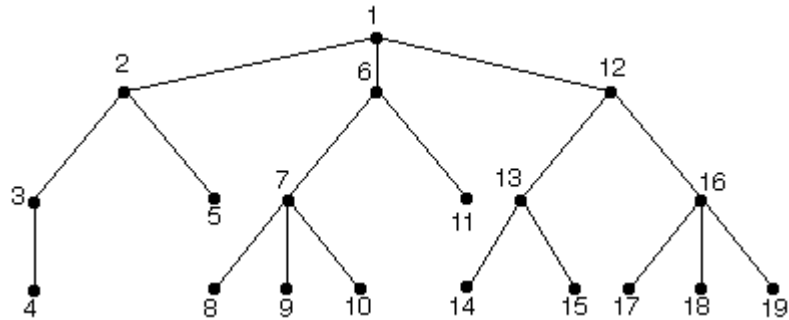


رسم 1 تمثيل شجرة البحث – العمق أولاً

الملاحظة: كل الأعداد في الرسم 1 تشير إلى تسلسل المرور في البحث.

بحث العرض أولاً

1. نعتبر المجموعة L قائمة العقد الأولية في المشكلة.
2. إذا كانت L فارغة، نصل إلى فشل من العقدة الأولى، في خلاف ذلك نأخذ العقدة n من L .
3. إذا كانت العقدة n هي حالة الهدف، نعاود الكرة من العقدة الأولية.
4. ما عدا ذلك تحذف الخطوة n من L ويتم اضافة كل اطفال ns إلى نهاية L . ضع علامة لكل طفل مع طريقه من العقدة الأولية. عد إلى 2.



رسم 2.

الملاحظة: كل الأعداد في الرسم 1 تشير إلى تسلسل المرور في البحث.

البحث الإرشادي

البحث الإرشادي هو المبدأ الذي :

- قد لا يعطي أفضل الحلول دائماً
 - لكن يضمن إيجاد حل جيد في الوقت المعقول.
 - بالكرار تزيد كفاءة.
 - مفيد في حل المشاكل الصعبة التي
 - o لا يمكن أن نحل بأي طريق آخر.
 - o حلول تأخذ وقتاً لانهاياً أو وقت طويل جداً للحساب.
- إن المثال الكلاسيكي لطرق البحث الإرشادية مشكلة البائع الجوال.

خوارزمية توليد واختبار طرق البحث الإرشادية

1. تولي الحل المحتمل والذي يُمكن أن يكون نقطة في فضاء المشكلة أو بداية الطريق من الحالة الأولية.
2. إختلِبْ هذا الحل المحتمل هل هو الحل الحقيقي بمقارنة الحالة التي تم الوصول إليها بمجموعة حالات الهدف.
3. إذا تبين انه هو الحل الحقيقي، نعود. بخلاف ذلك نكرر من 1.

هذه الطريقة أساس بحث العمق اولا وتعتبر كحلول كاملة يجب أن تحدد قبل الإختبار. ه ذه الطريقة تدعى طريقة المتحف البريطاني، حيث يتم الهُجْث عن معروض بشكل عشوائي. البحث الإرشادي مطلوب لتضييق البحث لنفترض انه لدينا مشكلة أربعة مكعبات ذات 6 أضلاع ، وكل جانب من المكعبات ملون بلُحْد الألوان الأربعة. إن المكعبات الأربعة موضوعة بجانب أحدها الآخر والمشكلة تكمن في ترتيبهم بحيث تكون الألوان المتوفرة الأربعة معروضة في كل جوانب المكعبات منطوية. المشكلة يُمكن فقط أن تحل إذا كان هناك على الأقل أربعة جوانب لونت في كل لون وعدد الخيارات المُجرب يُمكن أن يُخفَض بِلِستعمال البحث الارشادي بشرط ان اللون الأكثر شهرة يُخفي بالمكعب المجاور.

تسلق التل

هنا طريقة التوليد والاختبار مدموجة بوظيفة إرشادية والتي تقيس قرب الوضعية الحالية إلى حالة الهدف.

1. قِيم الحالة الأولية إذا هي حالة الهدف فنصل للنهاية، بخلاف ذلك تكون الوضعية الحالية حالة أولية.
2. نحدد مشغل جديد لهذه الحالة ونؤلّد حالة جديدة.
3. قِيم الحالة الجديدة

0 إذا ه ي أقرب لحالة الهدف من الحالة الحالية نقوم باعتمادها كحالة وضعية حالية

0 إذا ه ي ابعد فنُهْمَلْ

4. إذا الوضعية الحالية حالة الهدف أو ليس من مشغل جدي متوفر، نتوَك. ما عدا ذلك إعادة من 2.

في حالة المكعبات الأربعة العامل الإرشادي المناسب هو مجموع الألوان المختلفة على كل من الجوانب الأربعة، وحالة الهدف هي 4*16 على كل جانب. إن مجموعة القواعد تختار ببساطة المكعب وتدوي به بمقدار 90 درجة. الترتيب البادئ يُمكن أن يكون محدداً أو عشوائياً.

التلدين المُقلّد

هذا المبدأ مختلف عن تسلق التل ، والفكرة أن تتضمّن مسح عام من المشهد لتجنب تسلق التلال الخاطئة. إن الفضاء الكامل مُستكشف أولياً وهذا يتفادى خطر وجود هضبة أو حافة ويجعل الإجراء أقل احساساً إلى نقطة البداية. هناك تغييران إضافيان؛ نختار تحقيق حد أدنى بدلاً من خلق الحدود العليا ونحن نستعمل دالة التعبير الموضوعية بدلاً من الإرشادية. يصبح واضحاً بأننا بلنجداد وادي بدلاً من تسلق تل. يُلْهي هذا العنوان من عملية تحمية المعادن وبعد ذلك تركها تبرد حتى نصل الى حالة طاقة، أقل ما يمكن، النهائية الثابتة. الإحتمال الذي يمكن ان

$$p = \exp^{-\Delta E/kT}$$

يجعل المعدن يقفز إلى مستوى طاقة أعلى مُعطى بالمعادلة : حيث أن k هو ثابت بولتزمان

Boltzmann. إن المعدل الذي بيرد عنده النظام يُدعى جدول التلدين. ΔE يُدعى التغير في قيمة الدالة الموضوعية و kT تدعى T نوع درجة الحرارة. لعمثال لهذا النوع من المشاكل يمكن اعتبار خوارزمية البائع الجوال. إن خوارزمية التلدين المُقلّد مستندة على العملية الطبيعية التي تحدث في علم المعادن حيث ان المعادن تسخن إلى درجات الحرارة العالية ومن ثَمَّ يتوَيّها. ان نسبة او معدل التبريد تؤثر على المنتج التام الصنع بشكل واضح. إذا كانت نسبة التبريد سريعة، مثل حالة سقي المعدن ، بحيث يتم تبريد المعدن المحمي بشكل كبير بسرعة برميّه بخزان كبير مملوء بالماء، فإن التركيب في درجات الحرارة العالية يستمر في درجة الحرارة المنخفضة والتركيب البلورية الكبيرة يختلف معدلها بالمقارنة مع التبريد البطيء كما في الهواء الطلق.

1. قِيم الحالة الأولية.

2. إذا هـي حالة الهدف تتوَكَّف فيما عدا ذلك نَجْعَلُ الوضعية الحالية حالة أولي ونتابع.
3. نعطي صفة المتغير الحالة الأفضل للوضعية الحالية
4. نحدد درجة حرارة المجموعة، T ، طبقاً لجدول التلدين
5. الإعادة

AE-- اختلاف بين قيم الوضعيات الحالية والجديدة

1. إذا هذه الحالة الجديدة حالة الهدف ننهي البحث
2. نُقَارَنُ ما عدا ذلك بالوضعية الحالية
3. إذا كانت نتيجة 2 أفضل، نعطي قيمة الوضع الحالي كالحالة الأفضل، ونجعل الحالية هي الحالة الجديدة.
4. إذا هـي ليست أفضل، نَجْعَلُ الوضعية الحالية بالإحتمال p . هذا يَتَضَمَّنُ توليد عدد عشوائي في المدى من 0 إلى 1 وقَارَنُهُ مع النصف، إذا هو أقل من النصف لا نَعْمَلُ شيء وإذا هو أكبر من النصف نَقْبَلُ هذه الحالة كحالية قادمة.
5. معدل T في جدول التلدين معتمد على عدد العقد في الشجرة
- نكرر حتى وجود حل أو انتهاء أوامر التشغيل
6. أعطاء الجواب : أفضل _ حالة

البحث الأفضل الأول

يعتبر البحث الأفضل الأول عبارة عن مزيج من بحثي العمق أولاً والعرض أولاً. طريقة العمق أولاً جيدة لأن يمكن الوصول للحل دون الحاجة لحساب كل العقد، وطريقة العرض أولاً أيضاً جيدة لأنه لا يُحَصَرُ في النهايات المسدودة. البحث الأفضل الأول يَسْمَحُ لنا اليَقْلُ بين الطرق، وهكذا نَكْسِبُ المنفعة من كلتا الطريقتين. في كل خطوة، يتم اختيار العقدة الأكثر وَاَعِدَة. إذا تم إختيار إحدى العقد والتي تولد العقد التي أقل منها وتعد بأنها محتملة لإختيار آخر في نفس المستوى وفي الواقع البحث يَتَغَيَّرُ من العمق إلى العرض. إذا كانت النتيجة وبناءاً على التحليل أقل من العقد السابقة التي لاتزال محفوظة لدينا وطريقة البحث تَرْجِعُ إلى أحفاد الإختيار والإيرادات الأولى، عندها يتم التراجع.

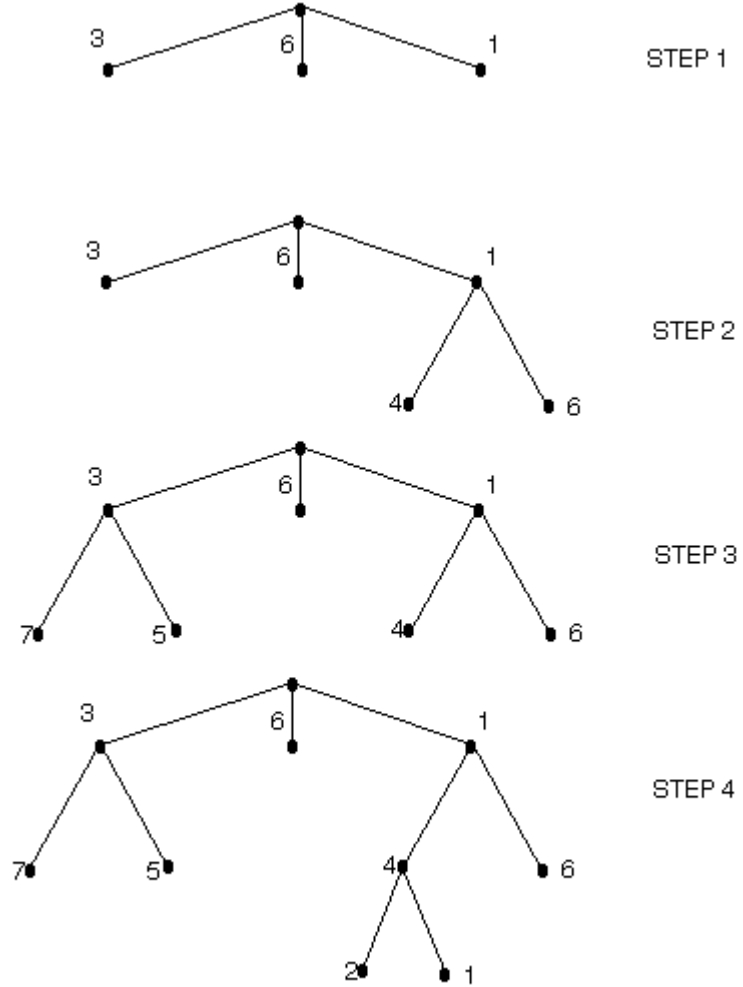
هذه العملية جداً مشابهة للاعتلاء الحاد، لكن في طريقة تسلق التلّ وعند اختيار الحركة يتم رفض العقد الأخرى ويعاد النظر فيها بينما في الأفضل أولاً هم مُوقَّرون لتأمين زيارة ثانية إذا يَحْدُثُ الطريق المسدود على الظاهر أفضل طريق. أيضاً أفضل حالة متوفرة مختارة في الأفضل أولاً وحتى ولو قيمتها أسوأ من قيمة العقدة المكتشفة عن طريق تسلق التلّ، وعليه يتوقف التقدّم في حال انه ليس هناك عقد وريث أفضل. أفضل خوارزمية للبحث الأفضل الأول تَتَضَمَّنُ مخطط OR و تتفادى أي تكرار للعقد وتُفَرِّضُ بأن كل عقدة لها وصلة أصل واحدة وذلك لإعطاء أفضل عقدة التي منها جاءت، وصلة إلى كل ورثتها. بهذه الطريقة إذا تم الوصول للعقدة الأفضل فيتم الانتقال للأسفل إلى الورثة. تَتَطَلَّبُ هذه الطريقة بلستعمال مخطط OR 2 من قوائم العقد.

يسمى الطابور الأولي للعقد مفتوح. إذا قِيمَ بالوظيفة الإرشادية لكن لم يتم التوسّع إلى الورثة. تكون العقد الأكثر وَاَعِدَة في الجبهة. يسمى الطابور بالمغلق إذا كانت العقد التي وُلِدَتْ والتي يجب أن تُخَزَّنَ لأن الوسم البياني (المخطط) يُسْتَعْمَلُ في التفضيل إلى شجرة.

البحث الارشادي يستخدم لايحاذ العقد الأكثر وَاَعِدَة، وتسمى الوظيفة الإرشادية f حيث أن f تقريب إلى f وتنبؤ من جزئين g و h حيث أن g كلفة الذهاب من الحالة الأولية إلى العقدة الحالية؛ g مُعْتَبَرُ ببساطة في هذا السياق الذي سيَكُونُ عدد الأقواس التي عبر كل منها ويُعْتَبَرُ من وزن الوحدة. h تخمين الكلفة الأولية يُصَبِّحُ من العقدة الحالية إلى حالة الهدف. إن الوظيفة f القيمة التقريبية أو تخمين يُصَبِّحان من الحالة الأولية إلى حالة الهدف. كلتا g و h متغيرات مقيّمة إيجابية. خوارزمية الأفضل الأولى هي أبسط شكل من خوارزمية A^* . من A^* نلاحظ بأن $f = g + h$ حيث أن g مقياس للوقت المطلوب للذهاب من العقدة الأولية إلى العقدة الحالية و h تخمين الوقت المطلوب للوصول إلى الحل من العقدة الحالية. هكذا f تخمين الوقت المستغرق للذهاب من العقدة الأولية إلى الحل. وللمساعدة نعتد الوقت المطلوب للذهاب من عقدة أخرى بحيث يَكُونُ ثابت وقيمتة 1.

أفضل خوارزمية بحث أولى:

1. البداية مع فتح حصّة الحالة الأولية
2. اعتبار أفضل عقدة على اساس انها مفتوحة
3. نولّد ورثته
4. لكل ورثه نعمل التالي:
 o إذا هو لم يُولّد قَبْلُ أَنْ يُقَيِّمَهُ نُضِيفُهُ لِفَتْحٍ وَنُسَجِّلُ وَالِدَهُ
 o إذا هو وُلِدَ قَبْلَ تَغْيِيرِ الْوَالِدِ، وَإِذَا هَذَا الطَّرِيقُ الْجَدِيدُ أَفْضَلُ، فِي هَذِهِ الْحَالَةِ نَجْدِدُ كَلْفَةَ الْوُصُولِ إِلَى أَيِّ عَقْدٍ وَرِثَةٍ
 5. إذا تم الوصول للهدف أو لا يوجد عَقْدٌ أَكْثَرَ فِي مِفْتَوحٍ، نصل للنهاية ، أو نَعُودُ فِيهَا عِدَا ذَلِكَ إِلَى 2.



All figures indicate "cost" of move

رسم 3. خوارزمية البحث الأولي

خوارزمية A*

أفضل بحث أول هو مُبَسَّط A*.

1. البداية مع فتح حصّة العَقْدِ الأولية.
2. لِتَقْطُ أَفْضَلَ عَقْدَةً عَلَى مِفْتَوحَةٍ مِثْلَ هَذِهِ الَّتِي $f = g + h$ أَقْلُ مَا يُمْكِنُ.
3. إِذَا أَفْضَلَ عَقْدَةً هَدَفٍ تَرَكْتَ وَتَرَجَعُ الطَّرِيقَ مِنَ الْأُولَى إِلَى أَفْضَلَ مَا عِدَا ذَلِكَ

4. أزل الأفضل من المفتوح وكل اطفال الأفضل، يعتبر كل بطريقه من العقدة الأولية.

الإنحطاط الرشيق للمقبولية

إذا 'h' يزيد في تقدير نادراً h بأكثر من d فإن خوارزمية A* ستجد الحل الذي كلفته d أعظم من الحل المثالي.

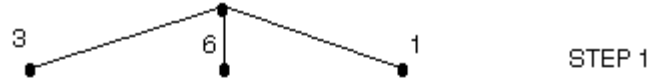
مخطط AND-OR

هذا المخطط مفيد لبعض المشاكل حيث :

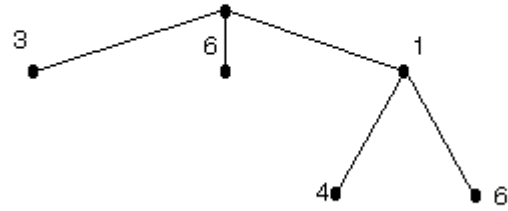
- يتضمن الحل انقسام المشكلة إلى المشاكل الأصغر.
 - نقوم بعدها بحل هذه المشاكل الأصغر.
- تتضمن هنا البدائل الفروع في أغلب الأحيان حيث البعض أو الكل يجب أن يرضي قبل امكانية التقدم. على سبيل المثال إذا أردت التعلّم لعزف فرانك Zappa المنفرد على القيثارة ، والذي يمكن أن نلاحظه (رسم).

(2.2.1)

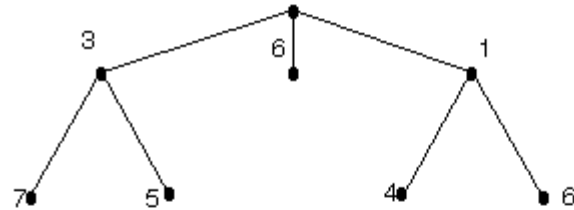
- إنسخه من القرص المدمج. أو
- إشتري كتاب قيثارة فرانك Zappa " وأقرأه من هناك.



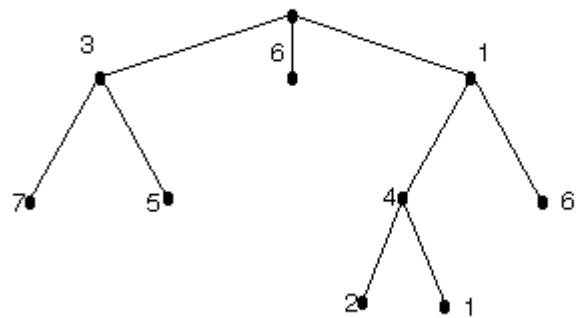
STEP 1



STEP 2



STEP 3



STEP 4

All figures indicate "cost" of move

رسم 4. مخطط AND-OR

ملاحظة : تم استعمال الأقواس للإشارة إلى ان واحدة أو أكثر من العقد يجب أن تكون محققة قبل انجاز العقدة الأصل . لإيجاد الحلول باستخدام مخطط AND-OR ، خوارزمية الافضل الأول تستعمل كقاعدة مع بعض التعديل لمعالجة مجموعة العقد المرتبطة بعامل AND.

خوارزمية AO*

1. بدء المخطط لبداء العقدة
2. عبر الرسم البياني متابعاً الطريق الحالي الذي يجمع العقد التي لحد الآن لم توسع أو تحل
3. إنقطة هذه العقد وسعها وإذا هي ليس لها ورثة رعو هذا عبث القيمة يحسب ما عدا ذلك فقط f لكل من الورثة.
4. إذا $0=f$ رؤس العقد بأنها حلت
5. غير قيمة f للعقدة المخلوقة حديثاً لعكس ورتبتها بالتوليد الخلفي.
6. استعمال الطرق الاكثر واعدة حيث هذا محتمل، وإذا العقدة محددة انها حلت وقم بتأشير العقدة الأصل انها حلت.
7. إذا عقدة البداية حلت أو تقيمها أعظم من العبث، توقف، ما عدا ذلك إعادة من 2.

تحليل وسائل - نهايات

- يسمح بالبحث بكل الاتجاهين جيئة وذهاباً.
- هذا يعني بأننا يمكن أن نحل أجزاء رئيسية من المشكلة أولاً وبعد ذلك نعود إلى المشاكل الأصغر عندما يجمع الحل النهائي.
- جي بي إس كان البرنامج الأول في الذكاء الاصطناعي لتنفيذ تحليل نهايات الوسائل.
- مخطط الرجل الآلي يعتبر مشكلة متقدمة Solver حيث يستخدم لحلها تحليل نهايات الوسائل والتقنيات الأخرى.

خوارزمية تحليل نهايات الوسائل:

1. كرر حتى الوصول للهدف أو لا إجراءات أكثر متوفرة:
 - o يصف الوضعية الحالية، حالة الهدف والاختلافات بين الإثنين.
 - o يستعمل الاختلاف يصف الإجراءات الذي سيصبح أقرب على أمل التهذيب.
 - o يستعمل الإجراءات ويجدد الوضعية الحالية.
2. إذا تم الوصول للهدف نعتبره نجاحاً ما عدا ذلك فشل.

رضاء القيد (تحقيق القيد)

- المشكلة العامة أن نجد الحل الذي يرضي (يحقق) مجموعة القيود.
- استعمال المخطط الارشادي ليس لنحتم المسافة إلى الهدف لكن لتقرير اي العقد تصلح لتوسيع العش.
- أمثلة هذه التقنية: مشكلة تصميم، تعريف رسوم بيانية، تخطيط طريق آلي والغاز.

الخوارزمية:

1. حدد القيود المتوفرة:
 - o رتّب كل الأشياء التي لها قيم ضمن الحل الكامل.
 - o كرر حتى كل الأشياء التي خصصت لها قيم صحيحة تكون صحيحة:
 - اختر جسماً ودعم قدر المستطاع مجموعة القيود التي تقدم إلى الجسم.
 - إذا مجموعة القيود مختلفة عن المجموعة السابقة قم بفتح كل الأشياء التي تشترك في هذه القيود.

- اعزل الجسم المحدد.
- 2. إذا تم اكتشاف إتحاد القيود في الاعلى نعرّف الحلّ كحل عودة.
- 3. إذا تم اكتشاف إتحاد القيود في الاعلى نعرّف التناقض لفشل عودة.

لماذا هذه المواضيع مهمة؟

بحث وشكل تمثيل المعرفة يعتبر القاعدة الاساسية للعديد من تقنيات الذكاء الاصطناعي.

سيتم سرد بضعة مؤشرات لايضاح استخدام طرق البحث المعينة في فصول هذا الكتاب.

تمثيل المعرفة

--البحث الأفضل الأول (A^*)، رضاء القيد وتحليل وسائل النهايات.

الاستنتاج المتردد

-- بحث العمق أولاً، بحث العرض أولاً وطرق رضاء القيد.

الاستنتاج المؤزّع

-- بحث الأفضل الأول (A^*) ورضاء القيد.

التخطيط

--بحث الأفضل الأول (A^*)، AO^* ، رضاء القيد وتحليل وسائل النهايات.

الفهم

-- رضاء القيد.

التعلّم

-- رضاء القيد، تحليل وسائل النهايات.

الحسن العام

-- رضاء القيد.

الرؤية

--العمق أولاً، العرض أولاً، البحث الارشادي، قلّد التلدين، رضاء القيد.

علم الإنسان الآلي

■ رضاء القيد ووسائل النهايات إستعمل في تخطيط الطرق الآلية.

برمجة الذكاء الاصطناعي باستخدام لغة البرمجة Prolog

سيتم في هذا الفصل سرد اساسيات برمجة الذكاء الاصطناعي prolog. التفاصيل الأخرى للغة ومن تقنيات برمجة prolog ستقدّم في الفصول التالية، حسب الحاجة لاعطاء الامثلة، ويُمكن الاستعانة بليّ كتاب prolog تخصصي آخر. بهدف الفائدة الاكبر سيتم إعطاء الكثير من الامثلة الخاصة ببرمجة الذكاء الاصطناعي بلغة Prolog. قد يجد القارئ لغة البرمجة Prolog لغة غريبة عن لغات البرمجة الأخرى المعروفة، ولكن من الضروري جدا لدارس الذكاء الاصطناعي التّعوّد على هذه اللغة. لذا انصح بإعطاء هذا الفصل الحظ الكافي من القراءة والفهم للحرص على الكثير من الممارسة.

برمجة الذكاء الاصطناعي

برامج الذكاء الاصطناعي، من حيث المبدأ، يُمكن أن تكون مكتوباً في أي لغة برمجة. على أية حال، كما هو الحال مع أي برمجة مهمة، هناك لغات التي لها الميزات التي تجعل الذكاء الاصطناعي يُبرمج بشكل سهل، واللغات الأخرى التي تجعل الأمر اصعب. هذا مفيد لنا للمرور بسرعة خلال البعض من خصائص برمجة الذكاء الاصطناعي التي تؤثر على إختيار اللغة.

دعم للبرمجة الإستطلاعية

لا تعتبر تقنيات هندسة البرامج مثل تكرير الخطوات والتطوير من المواصفات الأساسية مفيدة لإغلب مشاكل برمجة الذكاء الاصطناعي. لا يمكن إعطاء مواصفات كاملة لبرنامج الذكاء الاصطناعي قبل بناء نموذج له على الأقل - حيث اننا لا نفهم طبيعة المشاكل بما فيه الكفاية. تعتبر برمجة الذكاء الاصطناعي إستطلاعية. لذا عندما تُطوّر برنامج نحن نسعى لإستكشاف طبيعة المشكلة والمجال، ونكتشف إستراتيجيات الحَلّ الجيدة لتنفيذ المطلوب. هذه ليست دعوة لنترك تقنيات هندسة البرامج الجيدة، فقط لاننا لا نستطيع الإعتماد عليهم، وبأننا نطلب البيئة التي تدعم الأساليب الأكثر إستطلاعية من تطوير البرنامج. ميراث لغة برمجة / نظام والتي تدعم الإستطلاعية تتضمن: القدرة على تطوير مفسري اللغة لحل كافة أصناف المشاكل؛ وبيئات تطوير تفاعلية - حيث ان المبرمجين يُمكن أن يختبرون بحرية أقسام صغيرة بشكل تفاعلي.

اللغات الرئيسية للذكاء الاصطناعي

إن لغات البرمجة الرئيسية التي تستعمل في الذكاء الاصطناعي هي لغتي Lisp و Prolog. كلاهما له الميزات التي تعتبر مناسبة لبرمجة الذكاء الاصطناعي، مثل الدعم لمعالجة القائمة، مقارنة النقوش، والإستطلاعية. كلتا اللغتين كثيراتي الإستعمال - Prolog خصوصاً في أوروبا واليابان، Lisp في الولايات المتحدة. هذا الإستعمال العريض ضمن الحقل سبب آخر لإختيار اللغة Prolog لتطبيقات الذكاء الاصطناعي. هذا النوع من اللغات مستعمل بشكل اكبر في تطوير النظم الخبيرة، بينما نجده عاجزاً في النظم الذكية، حيث اننا بحاجة لحسابات كثيرة ومعقدة. كذلك وبما ان مدخلتنا في النظم الذكية هي صور (جرافيك) وهي بحاجة لمعالجة كبيرة، فإننا بحاجة هنا لإستعمال لغة أكثر علمية مثل فئات اللغة C, C++, Visual C او Pascal الخ. لغة البرمجة Lisp يُمكن أن تُعتبر كجذع البرمجة الوظيفية، طوّرت في أواخر الخمسينات ومستندة على تعاريف الوظيفية. بالرغم من نظرة استصغار من قبل أصولي البرمجة الوظيفيين، تبقى Lisp لغة برمجة كثيرة الإستعمال وقوية ومفيدة. تستعمل اللغة القائمة كتمثيلها الأساسي للتركييب وبرامج البيانات (تعاريف وظيفية)، ويؤدّ تشكيلة واسعة من بنى وتركييب في الوظائف لمعالجة القوائم. (في الحقيقة اللغة تؤيد معالجة قائمة). على خلاف اللغات الوظيفية الصافية، تركييب البيانات قد تُعدّل - ذلك محتمل نظرياً ومع انه مكروه لكن في الممارسة مفيدة جداً أحياناً! بالطبع، هذا ما زال محتمل لإستعمال اللغة على نحو وظيفي تماماً، وبشكل اختياري. إستعمال القوائم لتمثيل البرامج تُسمَح بكتابة مفسري اللغة بسهولة، حيث أن هناك إمتياز صَغير بين prolog والبيانات. تُسمَح القوائم ايضاً معالجة تركييب الرموز المعقدة بسهولة (كذلك تُمكن معرفة الذكاء الاصطناعي)، كذلك الامر بالنسبة لمقارنة النقوش والانماط يُمكن أن تكون مكتوبة بسهولة لمُجّارة تركييب القائمة.

لغة البرمجة Prolog لغة مستندة على المنطق. (Prolog = PROgramming in LOGic). عملياً، هي مستندة على الحساب الهندسي. كتابة برامج Prolog بسيطة ومشابهة لكتابة البيانات في الحساب المسند. عند كتابة برنامج prolog كبير (حيث اصبح لدينا معرفة اساسية بتركييب وادوات التحكم وآلية المطابقة في لغة البرمجة Prolog) من الأفضل التوقف عن التعامل مع البرنامج كبيانات في المنطق. كما في أي لغة برمجة أخرى، في النهاية يجب أن نعي تماماً ماهو المطلوب من البرنامج، وليس فقط معنى تركييب البيانات. عند كتابة برنامج بلغة البرمجة Prolog يجب أن يؤخذ بعين الاعتبار انها لغة برمجة اشتقّت من المنطق وإثبات النظرية، لكن مع قوفا الرئيسية نجد ان تركييب بياناتها مرنة جداً، كذلك تركييب قوفا لمطابقة الانماط، وبنى جيدة في إستراتيجية البحث.

أساسيات لغة البرمجة Prolog

أي برنامج prolog يشتمل مجموعة من الحقائق و مجموعة من القواعد. ليس هناك اي ضرورة لتحديد نوع المتغيرات في بداية البرنامج او تهيئة بدايات البرنامج كما في اللغات الاخرى. فقط بعض الحقائق والقواعد.

بعض الحقائق في prolog:

```
lectures(alison, ai).  
lectures(john, databases).  
female(alison).
```

```
age(alison, 29). [Oh, OK, that was last year..]
office(alison, s134).
animal(lion).
animal(sparrow).
has_feathers(sparrow).
```

الحقائق تَشمَلُها:

- اسم مسند (أو دالة) مثل المحاضرات، أنثى، مكتب...الخ. هذه الاسماء يَجِبُ أَنْ تَبْدَأَ بحرف صغير.
- صفر أو حجج بقيم، مثل 3 ai , alison , و s134.

مُلاحظة تلك الحقائق (القواعد، والأسئلة) يَجِبُ أَنْ تَنْتَهِى بنقطة.

بعض قواعد prolog:

```
bird(X) :-
    animal(X),
    has_feathers(X).

grandparent(X, Y) :-
    parent(X, Z),
    parent(Z, Y).
```

هنا ممكن ان نقول ان مشغل prolog :- 'يعني' اداة الشرط إذا"، بينما "،"، "يُمْكِنُ أَنْ نَقُولَ كمعنى" اداة العطف و". لذا القاعدة الأولى نَقُولُ بأنّ "X طير إذا X حيوان و X عِنْدَهُ ريش"، بينما القاعدة الثانية نَقُولُ بأنّ "Y,X جَدّ إذا Z,X والد و Y,Z والد". كُلّ الحجج تَبْدَأُ بحرف كبير (مثل X,Y) متغيّرات. (مُلاحظة تلك المتغيّرات لَمْ تُعالَج بنفس الطّريقة في لغات البرمجة التقليدية - على سبيل المثال، ليس من الضروري أن يملكوا قِيَم). أي ثابت يَجِبُ أَنْ لا يَبْدَأَ بحرف كبير ما عدا بأنه سَيَعْتَبَرُ متغيّر. لذا، أخذاً بعين الاعتبار حقيقة ان (Scotland, Edinburgh) كلتا الحجج سَتُعْتَبَرُ متغيّرات غير مقيّدة لأنها تَبْدَأُ بحرف كبير. تشغيل برنامج prolog يَتَضَمَّنُ أسئلة Prolog (بَعْدَ أَنْ تم بحمليّ مجموعتي الحقائق والقواعد). على سبيل المثال، أنت يُمْكِنُ أَنْ تَسْأَلَ:

```
?- lectures(alison, ai).
prolog يَعطِي الجواب "نعم" و إذا سألنا سلسلة من الأسئلة نحن نحصل على:
?- lectures(alison, ai).
yes
?- lectures(alison, databases).
no
```

الأسئلة يُمْكِنُ أَنْ تحتوي على متغيّرات، التي قَدْ تَحْصَلُ على قِيَمٍ معيَّنة، عندما prolog يَقوم بالإجابة على السؤال. Prolog سَيَعْرِضُ قيم نتائج كُلّ المتغيّرات في الاسئلة. لذا نحن سنحصل على:

```
?- lectures(alison, Course).

Course = ai
```

نحن يُمْكِنُ أَنْ نَسْأَلَ نفس السؤال بطريقة اخرى، مثل:

```
?- lectures(Someone, ai).

Someone = alison
```

نلاحظ ان فصل وشخص ما هي متغيرات يُمكنُ أَنْ تَأْخُذَ قِيَمَ من أي نوع.
نحن يُمكنُ أَنْ نكتشفَ من يحاضر ماذا بالسؤال:

?- lectures(Someone, Something) .

Someone = alison

Something = ai ;

Someone = john

Something = databases ;

no

بطباعة الفارزة المنقوطة بعد طباعة المجموعة الاولى من العناصر في برنامج Prolog ، نحن يُمكنُ أَنْ نرى إذا هناك أي عناصر اخرى محتملة. يَمُرُّ Prolog بكلِّ حقائقه وقواعده بشكل منظم ويُحاولُ إيجاد كُلِّ الطرق التي يُمكنُ أَنْ ترتبط المتغيرات بالقيم المعينة لكي تكون نتيجة الإستفسار الأولي مرضيَّة. على أية حال، كمثال عن كيفية استخدام القواعد ، يفترضُ بأننا نَسألُ السؤالَ التالي:

?- bird(B) .

ونحن عِنْدنا الحقائق والقاعدة:

animal(lion) .

animal(sparrow) .

has_feathers(sparrow) .

bird(X) :-

animal(X) ,

has_feathers(X) .

Prolog سيَطي النتيجة التالية :

B = sparrow.

يُقارن برنامج Prolog bird(B) مع رأس القاعدة (bird(X))، ويضع أسئلة جديدة، أولاً، animal(B) وبعد ذلك has_feathers(B) . Animal(B) يُمكنُ أَنْ يَكُونُ مقبول بملزم B إلى lion . على أية حال، يمكن القول ان :
has_feathers(lion) ليس صحيح. عليه يقوم برنامج Prolog بالتراجع ويُحاولُ من جديد B = sparrow. (عصفور) وهذا صحيح، ويعتبر حَلَّ محتمل.

مصطلحات Prolog وخاصة التوابع والتوحيد

في هذا القسم سنَتلقي نظرة أقرب على النحو والمعاني في برامج prolog.

النحو و تراكيب البيانات الأساسي

رَأيْنَا كيفَ برامج prolog تَشْمَلُ الحقائق والقواعد والأسئلة. تُعلنُ الحقائقُ الأشياء التي هي حقيقة دائماً. تُعلنُ القواعدُ الأشياء التي هي صحيحة بالإعتماد على بعض الشروط. البرنامج هنا يتألف من رأس و جسم. على سبيل المثال القاعدة "b(X), c(X) :- a(X) لها رأس "a(X)" وجسم "b(X), c(X)". يكون الرأس حقيقي إذا كانت كل الأهداف في الجسم صحيحة. (الحقائق يُمكنُ أَنْ تتعامل كقواعد بدون جسم). الحقائق و/ أو القواعد يمكن أن يُستعملوا لتعريف العلاقات (أو الاسناد). كما يُمكنُ أَنْ يَكُونُ لَهُ أعدادُ مختلفة من الحجج. على سبيل المثال، lectures/2 هي العلاقة الثنائية لـ lectures ، وقد تُعرَفُ بالحقائق الثلاث:

lectures(alison, ai), tt lectures(phil, databases), and lectures(gilbert, hci).

الأسئلة تُستعمل للاكتشاف فيما إذا الشيء حقيقي (وما هو العنصر المتغير المرتبط الذي سيحله صريح). الحقائق والقواعد والأسئلة كلها بنود prolog ويجب أن تنتهي بنقطة.

التعبير هو بنية تراكيب البيانات الأساسي في Prolog. التعبيرات في Prolog تتضمن:

مثل 102, 'Alison Cawsey', x, -. هذه تمثل كيانات وحيدة معينة عموماً ولا يمكن أن يفصل إلى أجزاء. تبدأ الذرات عادة بحروف صغيرة، لكن الرموز يمكن أيضاً أن يستعملوا إذا هم مقتبسون. رموز (مثل '``') وسلاسل الرموز (ومثال على ذلك: -، -، -) أيضاً تعتبر ذرات.

الأعداد:

مثل 29، 1.2، 3000، -2.

المتغيرات:

مثل X, Person, _Var. المتغيرات تبدأ أما مع حرف كبير أو بتأ برمز _.

الأجسام المنظمة:

مثل:

```
book(title(lord_of_the_rings), author(tolkien))
bicycle(owner(alison), parts(gears(number(18), type(shimano))))
```

هذه تتضمن عناصر (ومثال على ذلك: -، book كتاب، bicycle دراجة)، وبعض الحجج. الحجج قد تكون تباعاً أي تعبیر prolog.

[ملاحظة: الفرات في prolog قد تُلغى أو امر تشغيل. مثل امر التشغيل الحسابي * و+. بعد ذلك يمكن أن يستعملون لتشكيل شروط prolog مثل 3+1. نحن يمكن أن نكتب عنصر تقليدي أو تركيب حجة مثل (3, 1)+، لكن من الأسهل بكثير كتابتها 3+1. على أية حال تلك التعبيرات الحسابية لن تُقيم مالم تُجبر تقيماً مع بنى في مسند prolog "is"، على سبيل المثال.]

التراكيب المعقدة مثل الواردة اعلاه مفيدة إذا اردنا نقل كامل لبعض المعلومات حول مجموعة اجسام مرتبطة بعلاقة ما، أو جسم له العديد من الأجزاء، على سبيل المثال. نحن يمكن أن نتوصل إلى أجزاء تعبیر معقد بمُجاراته مع إستفسار (سيتم مناقشتها لاحقاً)، إذا كان عندنا برنامج Prolog كالتالي:

```
book(title(lord_of_the_rings), author(tolkien)).
```

```
has_famous_author(Title) :-
    book(title(Title), author(Author)),
    famous(Author).
```

ولدينا الإستفسار التالي:

```
?- has_famous_author(lord_of_the_rings).
```

Prolog يجاري اولاً الإستفسار مع رأس القاعدة، بحيث يقوم بربط Title مع lord_of_the_rings. ثم يطبق:

```
book(title(lord_of_the_rings), author(Author)).
```

هذه ستطابق بالحقيقة الأولى، و Author ستربط ب tolkien. ثم يقوم بتطبيق famous(tolkien)، الذي يجاري الحقيقة الثانية، مما يؤدي ببرنامج Prolog ان يعطي نتيجة نعم.

نلاحظ هنا كيف إستعمل Prolog مطابقة النماذج للتوصل إلى واحد من مجموعة معقدة من حقائق Prolog: (The author). سنناقش المطابقة بشكل مباشر أكثر لاحقاً.

كما رأينا، Prolog يحاول إثبات (اعطاء اجابة) الإستفسار بالبحث عن الحقائق التي تجاري ذلك الإستفسار، أو القواعد التي تجاري رؤوس الإستفسار والتي تملك اجسام يمكن أن يُثبت مطابقتها. يطابق prolog التعبيرات والشروط بطريقة حاسمة.

رَأَيْنَا كَيْفَ تَعَابِيرَ بَسِيطَةً مِثْلَ lectures(alison,Course) يُمكنُ أَنْ تُجَارَى إِلَى الْحَقَائِقِ مِثْلَ lectures(alison, ai) وصولاً إِلَى نَتِيجَةٍ مِثْلَ Course=ai. فِي الْقِسْمِ أَعْلَاهِ رَأَيْنَا أَمْثَلَةً أَكْثَرَ تَعَقُّبًا، حَيْثُ يَوْجَدُ تَعَابِيرُ بِالْحَجَجِ الْمَعْقَدَةِ تَمَّ مِطَابَقَتُهَا بِبَعْضِهَا الْبَعْضُ

```
book(title(Name), author(Author))
```

مِنَظَرَةٍ لـ:

```
book(title(lord_of_the_rings), author(tolkien))
```

مَعَ الْعُنَاصِرِ:

```
Name = lord_of_the_rings
Author = tolkien
```

يَجَارِي Prolog التَّعَابِيرَ عَلَى نَحْوِ هَيْكَلِي تَمَامًا، بِدُونِ تَقْيِيمِ التَّعَابِيرِ:

```
?- 1+2 = 3
no
```

[نَاحِظْ: فِي prolog "==" تَعْنِي "مِطَابَقَةٌ". نَحْنُ يُمكنُ أَنْ نَحْتَبِرَ الْمِطَابَقَةَ بِطِبَاعَةِ الْإِسْتَفْسَارَاتِ مِثْلَ الْوَارِدِ
أَعْلَاهِ.]
بِنَفْسِ الطَّرِيقَةِ التَّعَابِيرَ التَّالِيَةَ لِنَ جَارَى، حَيْثُ إِنَّ التَّعْبِيرَانِ تَرَكَيبُ مُخْتَلَفَةٌ:

```
?- X + 2 = 3 * Y.
no
```

بَيْنَمَا، تَتَمَّ الْمَجَارَةُ التَّالِيَةُ حَيْثُ لَدَيْهِمْ نَفْسُ التَّرْكِيبِ:

```
?- X+Y = 1+2.
X = 1
Y = 2
```

```
?- 1+Y = X + 3.
X = 1
Y = 3
```

مُلاحِظَةٌ: يُمكنُ أَنْ يَكُونَ هُنَاكَ مُتَغَيِّرَاتٌ فِي كُلَا الْجَانِبَيْنِ مِنَ التَّعَابِيرِ الْمَجَارَاةِ، كَمَا فِي الْمِثَالِ الثَّانِي أَعْلَاهِ.
كَأَمْثَلَةٍ غَيْرِ حِسَابِيَّةٍ، نَوْرِدُ:

```
?- lectures(X, ai) = lectures(alison, Y).
X = alison
Y = ai
```

```
?- book(title(X), author(Y)) = book(Z, author(tolkien)).
Z = title(X)
Y = tolkien
```

(فِي الْحَقِيقَةِ، فِي prolog نَحْنُ نَحْصِلُ عَلَى شَيْءٍ مَرَكَّبًا أَكْثَرَ قَلِيلًا رَدًّا عَلَى الْإِسْتَفْسَارِ أَعْلَاهِ، مَعَ Z = title(_0) and X = _0، بَيْنَمَا يَخْلُقُ prolog مُتَغَيِّرَاتِهِ الدَّاخِلِيَّةَ الْخَاصَّةَ. هَذِهِ مِكَافَأَةٌ إِلَى الْعُنَاصِرِ الَّتِي أُعْطِيَ أَعْلَاهِ.)
بِالطَّبْعِ، prolog عَادَةً سَيَقُومُ بِالْكَثِيرِ مِنَ الْمِطَابَقَاتِ بِالتَّسْلُسِ، كَمَا يُحَاوَلُ إِثْبَاتُ الْأَهْدَافِ الْفَرَعِيَّةِ فِي الْقَاعِدَةِ. ثُمَّ يَحْتَاجُ لِإِسْتِعْمَالِ الْعُنَاصِرِ الْمَتَغَيِّرَةِ الَّتِي تَمَّ الْحَصُولُ عَلَيْهَا بِالْمِطَابَقَةِ حَتَّى حِينَمَا يَقُومُ بِالْمِطَابَقَةِ الْتَّالِيَةِ. عَلَيْهِ يَمْكُنُ الْقَوْلُ:

```
?- X+Y = 1+5, X=Y.
```

no

[ملاحظة: نحن يُمكنُ أَنْ نَقُومَ بِجِدَّةِ إِسْتَفْسَارَاتٍ فِي السَّطَرِ الْوَاحِدِ، مَعَ فَاصِلَةٍ فِي الْوَسْطِ ، مِثْلَ فِي جِسْمِ الْقَاعِدَةِ فِي Prolog.]
X و Y نَحْصُلُ عَلَى قِيَمِ 1 5 عَلَى التَّوَالِي فِي الْمَجَارَاةِ الْأُولَى ($X+Y = 1+5$)، لَذا الْمَجَارَاةُ الثَّانِيَةُ عَمَلِيًّا $5=1$ ،
هِيَ الَّتِي تُوْدِي لِلْفَشْلِ. عَلَى أَيْةِ حَالٍ، الْمَجَارَاةُ الثَّالِيَةُ سَتُنْجَحُ:

```
?- X+Y = 1+5, Z = X.  
Z = 1  
Y = 5  
X = 1
```

```
?- book(author(X)) = book(author(tolkien)),  
famous(X) = famous(tolkien).  
X = tolkien
```

```
?- X=Y, Y = alison.  
X = alison  
Y = alison
```

نُلاحظُ انَّ إِنْثَانٍ مِنَ الْمَتَغْيِرَاتِ الْغَيْرِ مَقْيَدَةٌ مُتَنَازِرَةٌ عَلَيْهِ حَالَمًا يُصْبِحُ وَاحِدٌ مِنْهُمْ مُطَابِقٌ إِلَى بَعْضِ التَّعْبِيرِ يُصْبِحُ
الْآخَرُ مُطَابِقٌ بِشَكْلِ آلِيٍّ إِلَى ذَلِكَ التَّعْبِيرِ.
إِنَّ الْخَوَازِمِيَّةَ لِعَمَلِيَّةِ الْمُطَابَقَةِ فِي prolog مُسْتَنَدَةٌ عَلَى خَوَازِمِيَّةِ التَّوْحِيدِ الْمُسْتَحْدَمَةِ لِلْإثْبَاتِ الْآلِيِّ لِلنَّظَرِيَّةِ.
شَرْطَانِ يَطَابِقَانِ إِذَا أُمْكُنُ اسْتِدَادَ قِيَمٍ لِلْمَتَغْيِرَاتِ بِطَرِيقَةٍ بِحَيْثُ أَنْ، إِذَا الْمَتَغْيِرَاتِ فِي كُلِّ الشَّرُوطِ اسْتَبْدَلَتْ بِمُسْتَدَاتِهَا،
التَّعْبِيرَانِ يُصْبِحَانِ مُمَاتِلَانِ.

التَّراجِع

نَبْحَثُ فِي هَذَا الْقِسْمِ كَيْفَ يُجِيبُ Prolog عَلَى الْإِسْتَفْسَارَاتِ (وَيُذِيرُ الْبَرَامِجَ). أُعْطِيَ إِسْتَفْسَارٌ لِلْإثْبَاتِ (جَوَابٌ)،
يَرَاجِعُ Prolog قَائِمَتَهُ مِنَ الْحَقَائِقِ وَالْقَوَاعِدِ، مِنَ الْأَعْلَى إِلَى الْأَسْفَلِ، يَبْحَثُ عَنْ رُؤُوسِ الْقَوَاعِدِ أَوْ الْحَقَائِقِ الَّتِي
تَطَابِقُ الْإِسْتَفْسَارَ. عِنْدَمَا يَجِدُ وَاحِدًا، يَخْزِنُ الْمَكَانَ الَّذِي وَصَلَ عِنْدَهُ فِي بَحْثِهِ فِي قَاعِدَةِ الْحَقِيقَةِ / الْقَاعِدَةِ ، إِذَنْ لَوْ أَنَّ
الْمَجَارَاةَ الْأُولَى هِيَ إِكْتِشَافَاتٌ غَيْرُ جَيِّدَةٍ، يُمَكِّنُ مُوَاصِلَةَ الْبَحْثِ عَنِ الْإِمْكَانِيَّاتِ الْآخَرَى.
لِنَقْتَرِضْ بِأَنَّهُ عِنْدَنَا الْحَقَائِقُ الثَّالِيَةُ:

```
bird(type(sparrow), name(steve)).  
bird(type(penguin), name(tweety)).  
bird(type(penguin), name(percyp)).
```

وَيُعْطِي الْإِسْتَفْسَارَ ?- bird(type(penguin), name(X)).
Prolog سَيُحَاوِلُ مُجَارَاةَ الْإِسْتَفْسَارِ أَوَّلًا بِالْحَقِيقَةِ الْأُولَى، لَكِنْ يَفْشَلُ لِأَنَّ الْعَصْفُورَ لَا يَطَابِقُ بِطَرِيقٍ. بَعْدَ ذَلِكَ
سَيُحَاوِلُ الْمُجَارَاةَ بِالْحَقِيقَةِ الثَّانِيَةِ، وَتَنْجَحُ مَعَ $tweety = X$. عَلَى أَيْةِ حَالٍ، هُوَ سَيَبْذُرُ مُؤَشِّرَ يَدِلٍ عَلَى الْمَكَانِ الَّذِي
وَصَلَ إِلَيْهِ، إِذَنْ لَوْ يَظْهَرُ أَنَّ الْإِسْتَفْسَارَ / الْهَدَفَ الْفَرْعِيَّ الْتَالِيَّ يُوْدِي لِلْفَشْلِ، سَيَعُودُ إِلَى الْمَوْقِعِ الْمُخْزَنِ قَبْلَ ذَلِكَ،
وَيَبْحَثُ عَنِ الْحُلُولِ الْآخَرَى ($percyp = X$).
لَذا، إِذَا كَانَ عِنْدَنَا حَقِيقَةٌ / قَاعِدَةٌ مِثْلُ:

```
animal(leo).  
animal(tweety).  
animal(percyp).  
animal(peter).  
has_feathers(percyp).  
has_feathers(peter).  
tame(peter).
```



```
bird(X) :- animal(X), has_feathers(X).
tame_bird(X) :- bird(X), tame(X).
```

إذا أعطى الإستفسارَ `bird(Bird)` ، فإن Prolog سيمر نزولاً بالحقائق والقواعد حتى الوصول إلى القاعدة الأولى، ويجاري `bird(Bird)` مع `bird(X)`. المتغير `Bird` يسند إلى المتغير `X`. بعد ذلك يحاول Prolog إثبات جسم القاعدة. الهدف الثانوي `animal(X)` يُنجح مع `X=leo` و يبقى Prolog المؤشر هكذا ليدل أين هو أصبح في التدقيق للـ `animal(X)` ، ويحاول بـ `has_feathers(leo)`. ينزل اسفلاً عبر الحقائق / القواعد للبحث عن تلك التي تجاري، لكن لا يجد أي شيء. يعود إلى المجارة الأخيرة، يحاول مجارة `(animal(X))`، حيث يبدأ من حيث تخلق عنها، ويقوم بمجارة `animal(tweety)`. لكن `has_feathers(tweety)` ما زال لا يجاري. لذا يعود ثانيةً ويجد `animal(perc)`. مجارة `has_feathers(perc)` ، لذا الموضوع بأكمله يُنجح، بـ `Bird=perc`.

الآن، إذا نعطى الإستفسارَ `tame_bird(B)` ، يُحصل تعقيد أكبر نوعاً ما. يُحاول Prolog أولاً أن يحقق `bird(X)` كما حصل أعلاه. لكن `tame(perc)` تصل إلى الفشل. لذا يجب أن يُحاول إلى إعادة تحقيق `bird(X)`. يقوم بذلك بالمحاولة الأولى إلى إعادة تحقيق `has_feathers(perc)` كذلك كان الهدف الأخير في المحاولة. لكن ليس هناك طريق آخر من تحقيقه. لذا يعود ثانيةً إلى `animal(X)`، لإيجاد `X = peter`. يُنجح `has_feathers(peter)`، لذا `bird(X)` يُمكن أن يُنجح الطريق الثاني مع `X = peter`. `tame(peter)` يُنجح، كذلك `so tame_bird(B)` مع `B = peter`.

في المثال أعلاه ، prolog فقط يتراجع خلال الحقائق. لكن عموماً هناك قد يكون أكثر من قاعدة واحدة التي يمكن لـ prolog أن يحاول، على سبيل المثال:

```
animal(perc).           % fact1
has_feathers(perc).     % fact2
penguin(tweety).        % fact3
penguin(peter).         % fact4
tame(peter).            % fact5

bird(X) :-               % rule1
    penguin(X).

bird(X) :-               % rule2
    animal(X),
    has_feathers(X).

tame_bird(X) :-          % rule3
    bird(X),
    tame(X).
```

أعطى إستفسار `bird(B)` ، prolog يستعمل القاعدة الأولى أولاً (والحقيقة الثالثة) وتُصبح `B=tweety`. إذن كان لا بُد أن يتراجع ويجاري `penguin(X)` بالحقيقة الرابعة لتُصبح `B = peter`. إذا هو يتراجع ثانيةً وهو سيستنفذ بنود البطريق لذا يجب أن يحاول مع قاعدة الطير الثانية، و(تستعمل الحقيقة الأولى والثانية) تُصبح `B = perc`. لذا، يجب ان نذكر بأن يُمكن أن تُطبع "؛" (أو فقرة على "قادمة") لخيار التراجع، مما يؤدي للتالي:

```
?- bird(B).
B = tweety ;
B = peter ;
B = perc ;
no
```

هذا قد (أو قد لا) يكون أوضح إذا رسمنا بعض الأشجار. بنية تراكيب الهدف/ الهدف الثانوي في Prolog تمثل كشجرة " and-or " .

وجهات النظر التصريحية والإجرائية للبرامج

في هذه النقطة من الخطأ التفكير بمعنى برامج Prolog. هناك عموماً طريقان للنظر إلى أي قاعدة في Prolog –أولاً كبيان تصريحي حول ماهو حقيقي، أو كبيان إجرائي حول كيف ننفذ شيئاً ما. إحدى الفوائد المُدَّعية لـ Prolog بأنها في أغلب الأحيان بؤمجة تماماً تصريحية. يجب ان نكون مطمئنين بخصوص كيفية اثبات الاشياء في Prolog. على أية حال، لأي برنامج معقد إلى حدّ معقول، كما ناقشنا في وقت سابق، يُصبح مهم في أغلب الأحيان التفكير بشأن كيف يُنفذ البرنامج. على سبيل المثال، طلب تنفيذ الهدف الثانوي قد يكون حرج بالنسبة إلى كفاءة البرنامج، والمتغيرات قد تُستعمل بطريقة بحيث لا تؤدي الى قراءة تصريحية بسيطة. يُسمح Prolog بالبيانات غير المنطقية أيضاً، مثل كتابة البيانات والمزاعم. عندما تكون هذه الاشياء مُتضمنة في برنامج يُصبح هذا البرنامج مهم لدرجة أكبر للتفكير بشأن الذي يمكن لـ Prolog ان يعمله. لنقول بشكل واضح ما الذي نغني بالقراءات التصريحية والإجرائية من البرامج، لنفترض انه عندنا قاعدة:

p :- q, r.

إن القراءة التصريحية: "p حقيقي إذا q حقيقي و r حقيقي". القراءة الإجرائية يُمكن أن تكون كالتالي "لحل المشكلة p، نحل مشكلة q أولاً وبعد ذلك نحل مشكلة r"، أو بدلاً عن ذلك "لتحقيق p، نحقق أولاً q وبعد ذلك r".

التكرار

تقريباً أي برنامج غير استنباطي من prolog يتضمّن مسندات تكراريّة – المسندات التي تنادي نفسها بنفسها. الفكرة الأساسية يجب أن تكون مألوفة من تعاريف الوظيفة التكرارية في اللغات الوظيفية. على أية حال، بما ان برنامج Prolog ليس مستند على تطبيق الوظيفة، الطريق التكراري للمسندات المَستعمل والمكتوب مختلف قليلاً. نفترض اننا نريد كتابة إجراء Prolog للتقرير سواء شخص ما سلف (اسلاف) شخص آخر. هذا له تعريف تكراري طبيعي. X هو سلف Y، إذا X والد Y أو Z والد Y و X سلف Z. هذا يُمكن أن يُكتب كالتالي:

```
ancestor(Person, Ancestor) :- % Rule 1: Base case
    parent(Person, Ancestor).

ancestor(Person, Ancestor) :- % Rule 2: Recursive case
    parent(Person, Parent),
    ancestor(Parent, Ancestor).
```

نلاحظ كيف ان الحالة الأساسية للتعريف التكراري قاعدة منفصلة، تتبّع الحالة التكرارية. كل المسندات التكراريّة يجب أن يكون لهم حالة أساسية، بخلاف ذلك أمّا يؤديوا الى الفشل او يبقون في تكرار إلى الأبد. إعتبر ما يحدث إذا عندنا الحقائق التالية أيضاً:

```
parent(alison, david). % fact 1
parent(alison, kathleen). % fact 2
parent(david, harold). % fact 3
parent(david, ida). % fact 4
parent(kathleen, john). % fact 5
```

ونحن نسأل:

?- ancestor(alison, harold).

Prolog سَيُطابق الإستفسارَ مع القاعدة 1، ويُحاول إثبات `parent(alison, harold)`. هذا سَيُفشل، لذا Prolog سَيُترجعُ ويُحاول القاعدة الثانية. `parent(alison, Parent)` يَنْجُحُ أولاً مع `Parent = david`، لذا Prolog يُحاولُ إثبات `ancestor(david, harold)`. يجاري هذا التعبير الجديد مع رأس القاعدة الأولى، وينجح يُثبت `parent(david, harold)`، مما يؤدي الى انجاح الإستفسار بشكل كامل.

إذا نحن نُحاولُ إثبات `ancestor(alison, john)` نَتَقَدُّ الامور شيئاً قليلاً حيث Prolog يَجِبُ أَنْ يَتَرَجَعَ. `parent(alison, david)` غير جيّد، حيث يجب انه غير قادر على إثبات السلف `(david, john)`. يعني ان البرنامج يَجِبُ أَنْ يَتَرَجَعَ، لإيجاد والد `(alison, kathleen)`، وبرهنة سلف `(kathleen, john)`.

من الحكمة ان لانفكر بصعوبة الذي يَحْدُثُ عندما يَتَرَجُعُ Prolog خلال تعاريف المسندات التكرارية المعقدة. التعاريف يُمكنُ أَنْ يَكُونُوا مفهوماً بسهولة في أغلب الأحيان بشكل تصريحي، ونحن يَجِبُ أَنْ نُنْعِي ماذا سَيَحْدُثُ، بدون الحاجة لَتَتَبُعُ كُلَّ شيءٍ بالتفصيل. إفتراض على سبيل المثال نُعطي الإستفسار التالي، وندفع للترجع (بطباعة الفارزة المنقوطة).

```
?- ancestor(alison, Ancestor).
Ancestor = david ;
Ancestor = kathleen ;
Ancestor = harold ;
Ancestor = ida ;
Ancestor = john ;
no
```

يَجِدُ Prolog كُلَّ الحقائق أولاً التي تجاري الأهداف الثانوية للحالة الأساسية (سلف `david = kathleen`). ثم يَجِدُ كُلَّ أسلاف `david (ida, harold)`، ثم كُلَّ أسلاف `kathleen (john)`. إذا أضفنا حقيقة ان والد `(harold, fred)` ثم سلف `fred =` سَيُعطي بعد السلف `ida =`، بينما Prolog حاول إيجاد كُلَّ أسلاف `harold`.

متابعة تنفيذ برامج Prolog

في هذه النقطة يَجِبُ أَنْ نُحاولُ إستعمال وسائل متابعة الأثر في Prolog، ورؤى إذا من المُمْكِنُ أَنْ نتابع الذي يَحْدُثُ كما في المثال اعلاه. وسائل الأثر قَدْ تَتفاوتُ بعض الشيء من تطبيق إلى تطبيق، لكن كُلُّها مستندة على نفس النموذج الأساسي، المعروف بنموذج الصندوق. أساساً، لأي إجراء `prolog (say, ancestor)` نحن يُمكنُ أَنْ نَجعل برنامج Prolog إخبارنا عندما يَدْعُو ذلك الإجراء، وعندما يَخْرُجُ (ينجح) ذلك الإجراء، وعندما يُحاولُ ثانية ذلك الإجراء (بسبب المتابعة الخلفية) وعندما يصل أخيراً ذلك الإجراء الى الفشل. أنت يُمكنُ أَنْ تُحدّدَ إجراء الإهتمام بوضع نقاط مراقبة، ولكن نحن الان سَنُنْظُرُ فقط الى النوع الأسهل للأثر، حيث هو يُخبرُك عن كُلِّ إجراء. في Prolog التالي يمكن اعطاء المثال التالي:

```
| ?- trace.
```

```
yes
```

```
| ?- ancestor(alison, Anc).
(1) 1 call: ancestor(alison,_0) ?
(2) 2 call: parent(alison,_0) ?
(2) 2 exit: parent(alison,david)
(1) 1 exit: ancestor(alison,david)
```

```
X = david ;
(1) 1 redo: ancestor(alison,_0) ?
(2) 2 redo: parent(alison,_0) ?
(2) 2 exit: parent(alison,kathleen)
(1) 1 exit: ancestor(alison,kathleen)
```

```
X = kathleen ;
```

```

(1) 1 redo: ancestor(alison,_0) ?
(2) 2 redo: parent(alison,_0) ?
(2) 2 fail: parent(alison,_0)
(3) 2 call: parent(alison,L103) ?
(3) 2 exit: parent(alison,david)
(4) 2 call: ancestor(david,_0) ?
(5) 3 call: parent(david,_0) ?
(5) 3 exit: parent(david,harold)
(4) 2 exit: ancestor(david,harold)
(1) 1 exit: ancestor(alison,harold)

```

X = harold ;

يُذَعو Prolog اولا السلف (alison , _ 0) (حيث أنَّ \ 0 اسمَ لمتغير داخلي في Prolog)، ويذَعو الهدف الثانوي، والد (alison , _ 0). هذا يتم بنجاح، و يُؤدّي الى الوصول الى نتيجة ان السلف (alison , _ 0) يصل للخروج مع السلف (alison , david). عندما يقوم Prolog بالتعقب الرجعي ، يُعيدُ Prolog مجارة السلف (alison , _ 0)، ويبدأ بإعادة مجارة الوالد (alison , _ 0)، الذي يؤدي الى الخروج بنتيجة مع الوالد (alison , kathleen). كذلك يؤدي الى خروج السلف (alison , _ 0). بتكرار التعقب الرجعي ثانية، يُحاولُ Prolog مجارة السلف (alison , _ 0). إعادة مجارة الوالد (alison , _ 0) تؤدي الى الفشل -حيث انه ، ليس هناك مزيد من الحقائق التي تجاري هذه المطابقة. يستعملُ السلفُ الثاني الذي يقوم ببناء والد (L103 , alison) (حيث أنَّ L103 متغير داخلي في Prolog). هذا يُنَجِّحُ، حيث، يستعملُ الهدفُ الثانوي الثاني مِنْ قاعدة 2 التي تُذَعو السلف (david , _ 0). هذا يُنَجِّحُ ، ويؤدي الى مخرج، مع الوالد (david , harold)، حيث ان النداء الثاني إلى السلف ادى الى الخروج، وكذلك الأول.

وسائل الأثر في Prolog، حسب ما رأينا في المثال أعلاه، تتفاوت بين التطبيقات. في النهاية علينا أن نتعرّف على خواص وسائل الأثر في Prolog للوصول الى افضل النتائج.

معالجة القائمة

حتى الآن نحن رأينا تراكيب البيانات من نوع مشغل / حجة. هذا يجعل الى حدّ ما الامور غير مرنة للعديد من التطبيقات. نريد أن يكونَ عندنا بعض القوائم أو المجموعات التي يُمكن أن نُضيفَ اليها او نحذف منها الينود حسب الرغبة. هذا يعني اننا بحاجة الى تراكيب القائمة. تراكيب القائمة الصحيحة في prolog تتضمن:

```
[1, 2, 3]
```

```
[alison, richard]
```

```
[1, 2, 3, "go"]
```

```
[1+2, [alison], widget(handle(2))]
```

```
[name=widget21, size=22, colour=blue]
```

```
[alison, [[david, [[harold], [ida]]], [kathleen, [[john]]]]]
```

```
[] (empty list)
```

من الصعب التفكير بسياق حيث يكون مفيداً، والقائمة الأخيرة قد تمثل بشكل أفضل لالتركيبة التقليدية في Prolog كشجرة تركيب/حجة ، مثل:

```
person(name(alison), parents(person(name(david), parents(...)),
person(...)))
```

على أية حال، عموماً قائمة Prolog قد تشمل شروطاً إعتباطية، من ضمن ذلك القوائم. كما في اللغات الوظيفية، نحن يمكن أن نختار رأس وذيل القائمة من خلال المُجَاراة. في Prolog، النمط الخاص [V1|V2] يجاري القائمة بمطابقة الرأس مع V1 والذيل مع V2:

```
?- [H|T] = [1,2,3,4].
```

```
H = 1
```

```
T = [2,3,4]
```

```
?- [alison|Rest] = [Person, richard, alan, jeff].
```

```
Person = alison
```

```
Rest = [richard, alan, jeff]
```

```
?- [person(X)|Rest] = [person(fred), person(joe)]
```

```
X = fred
```

```
Rest = [person(joe)]
```

```
?- [H|T] = [a]
```

```
H = a
```

```
T = []
```

```
?- [H|T] = []
```

```
no
```

```
?- [a,b|C] = [a,b,c,d]
```

```
C = [c,d]
```

```
?- [a,b|[c|D]] = [a,b,c,d]
```

```
D = [d]
```

من المثاليين الأخيرين، نلاحظ ان الانماط التي تتضمن ``|`` ليس من الضروري أن تكون بسيطة [رأس | ذيل] أنماط – عدة بنود يمكن ان تتبع هذا الرمز ``|`` ، ويكون هذا ضمن قائمة ثانوية. الهند بعد ``|`` "يجب أن يجاري بقية بنود القائمة دائماً".

نحن يمكن أن نبدأ بكتابة بعض عناصر ومسندات معالجة القائمة. نفترض اننا نريد كتابة عضو مسند، للإكتشاف إذا عنصر ما موجود في القائمة المُعطاة. يعتبر بند ما عضو في قائمة إذا كان هو رأس القائمة، أو إذا كان عضو في ذيل القائمة، لذا:

```
member(Item, [Item|_]). % It's a member if it matches the head.
```

```
member(Item, [_|Tail]) :-
```

```
member(Item, Tail). % or if it's a member of the tail.
```

نلاحظ إستعمال المتغيرات الغير مسماة ``_``. مما يسهل إستعمال هذه الاسماء المتغيرة للأشياء التي لا نهتم بها. نحن يمكن أن نستعمل العضو المسند للتدقيق إذا شيء ما هو عضو في قائمة، كذلك نستخدمه لتوليد أعضاء جدد في القائمة، أو حتى لتوليد القوائم التي هو عضو بها:

```
?- member(a, [b, a, c]).
```

```
yes
```

```
?- member(I, [b, a, c]).
```

```
I = b ;
```

```
I = a ;
```

```

I = c ;
no

?- member(a, L) .

L = [a|_16] ;
L = [_15,a|_20] ;
L = [_15,_19,a|_24]
yes

```

فكر كيف يتراجع Prolog للحصول على حلول مختلفة.

العضو لا يعمل فقط مع القوائم البسيطة. بل يمكن أن تكون قوائم تشمل شروط معقدة. هذا سينجح إذا طبقت الحجة الأولى بند ما في القائمة المعطاة في الحجة الثانية. أكثر المسندات في Prolog على هذا الشكل - يُدققون إذا الأشياء متطابقة، وليس إذا الأشياء نفسها، وتكون نتيجة المُجَارة بأن تربط بعض المتغيرات ببعضها البعض.

```

?- member(book(tolkien, X), [book(cawsey, explanation), book(tolkien,
lord_of_rings)]).
X = lord_of_rings

?- member(book(X, Y), [book(cawsey, explanation), book(tolkien,
lord_of_rings)]).

X = cawsey
Y = explanation;

X = tolkien
Y = lord_of_rings

```

وهكذا يضاف مسندا آخر:

```

append([], List, List).
append([H|T], List, [H|New]) :-
    append(T, List, New).

```

اضيفت قائمة فارغة فقط لإدراج تلك القائمة. تم اضافة قائمتان تشتملان سوية قائمة جديدة مع الرأس، رأس القائمة الأولى، وتتعبان نتيجة اضافة ذيل القائمة الأولى إلى الثانية. أي مثال مُتَّبَع يعطي:

```

?- append([1,2], [3], L).
(1) 1 call: append([1,2],[3],_9) ?
(2) 2 call: append([2],[3],_27) ?
(3) 3 call: append([], [3],_34) ?
(3) 3 exit: append([], [3], [3])
(2) 2 exit: append([2], [3], [2,3])
(1) 1 exit: append([1,2], [3], [1,2,3])

L = [1,2,3]

```

لمعالجة القائمة التكرارية لمسندات مثل العضو وإستراتيجية الاضافة الأساسية، يجب أن نكتب الحالة الأساسية التي تحل بعض النسخ البسيطة للمشكلة، كذلك الحالة التكرارية التي تبقى النتيجة حقيقة، وحيث يثرب كل نداء تكراري أكثر نحو الحالة الأساسية. في حالة الاضافة، التركيب الذي سيرجع يُعزّز في رأس البند ([H|New]). الإستراتيجية المألوفة الأخرى هي عندما يتم كتابة مسندات معالجة القائمة باستخدام متغير إضافي لإبقاء بعض النتائج بعيدة شئ ما. مثال واضح على هذا هو مسند لإيجاد حد أعظمي للقيم في قائمة ما:

```

max([], MaxSoFar, MaxSoFar).

```

```

max([Number|Rest], MaxSoFar, Max) :-

```

```
Number > MaxSoFar,
max(Rest, Number, Max).
```

```
max([Number|Rest], MaxSoFar, Max) :-
    Number <= MaxSoFar,
    max(Rest, MaxSoFar, Max).
```

هذا يُمكنُ أَنْ يُدْعَى كالتالي:

```
?- max([1,6,23,21], 0, Max).
```

(يفترضُ ان الأعداد التي ضمننت في القائمة أكبر مِنْ 0).
 لإسناداتٍ مثل هذه من الأسهل إعطاء حساب أكثر إجرائيةً من الذي يَحْدُثُ (الذي ليسَ مختلف كثيراً عن الذي يَحْدُثُ
 في برنامج باسكال). لإيجاد القيمة العظمى في قائمة، نفترض حدَّ أقصى للقيمة ، ونقارن مع العنصر التالي إذا كان
 أكبر مِنْ الاعظمي المفترض حتى الآن نجعلُ هذا العنصر اعظمي جديد ، نحاولُ مع العناصر الباقية. إذا العنصر
 التالي ليسَ أكبرَ فقط نَستمرُّ إلى العنصر الذي يليه في القائمة. عندما نصلُ إلى نهاية القائمة فإن العنصر الاعظمي
 المفترض حتى الان يَكُونُ حدَّ أعظمي لعناصر القائمة.
 نعطي هنا بعض الامثلة عن مسندات معالجة القائمة:

```
% reverse(+List,+SoFar,-ReversedList)
% Should succeed with ReversedList bound to reversed List (appended
to
% initial value of SoFar). Normally called with SoFar=[]/

reverse([], SoFar, SoFar).
reverse([H|T], SoFar, Final) :-
    reverse(T, [H|SoFar], Final).

% delete(+Item, +List, ?NewList)
% Should succeed with NewList bound to List but with all items
matching
% Item removed.

delete(_, [], []).
delete(Item, [Item|Rest], NewRest) :-
    delete(Item, Rest, NewRest).
delete(Item, [First|Rest], [First|NewRest]) :-
    not First = Item,
    delete(Item, Rest, NewRest).

% replace(+Item, +NewItem, +List, -NewList)
% Should succeed with NewList bound to List but with all items
% matching Item replaced by NewItem

replace(_,_, [], []).
replace(Item, New, [Item|Rest], [New|NewRest]) :-
    replace(Item, New, Rest, NewRest).
replace(Item, New, [First|Rest], [Item|NewRest]) :-
    not First = Item,
    replace(Item, New, Rest, NewRest).
```

لاحظُ إستعمالَ ``Not " للتدقيق إذا شيء ما ليسَ صحيح. بعضُ مبرمجي Prologs يُفضلُ إستعمالَ + بدلاً مِنْ
 Not. إنَّ السببَ هو انها ليسَ Not المنطقي - Prolog فقط يَدَقِّقُ لمعرفة إذا كان بالامكان اثبات شيء ما، وإذا لا
 يستطيعُ فإن ``Not " تؤدي الى النجاح (بطريقة اخرى ممكن القول انه `` أنا ليسَ متأكدٌ لكني لا أستطيعُ إثبات ذلك

("). البعض الآخر من مبرمجي Prologs لا يَحْبُونَ إستعمال `` Not " لتمثيل هذه الفكرة (المدعوة باسم `` إنكار كفشل ").

المواضيع الباقية

ما غَطَيْنَاهُ حتى الان يعتبر مقدمة للتعريف بلغة Prolog, ولم نتطرق لكافة مواضيع اللغة، حيث ان تفاصيل هذه اللغة ليست موضوع هذا الكتاب. لكن الحاجة لمعرفة بعض المفاهيم عنها، ولضرورة سرد بعض الامثلة في هذا الكتاب في مختلف اقسام الذكاء الاصطناعي باستخدام هذه اللغة، جعل من الضروري اعطاء هذه اللمحة. لمعرفة تفاصيل اكبر عن لغة Prolog يمكن الاستعانة بكتاب متخصص بهذه اللغة. كما يمكن مستقبلا ان يكون هناك كتاب متخصص بهذه اللغة ضمن مجموعة الكتب المخطط لاصدارها، والتي تغطي كافة فروع الذكاء الاصطناعي وادواته.

تمثيل المعرفة

مادة التَّمثِيل

لنحدد أولاً أنواع المعرفة التي يمكن أن تُمَثَّل في أنظمة الذكاء الاصطناعي:

الأجسام

-- حقائق حول الأجسام في مجالنا الحيائي. ومثال على ذلك: - آلات وتريه كالقيثارة ، آلات نحاسية كالأبواق.

الأحداث

-- عمل او حدث يقع في حياتنا. ومثال على ذلك: - زيد عزف قطعة موسيقية على آلة العود.

الأداء

-- أي سلوك مثل العزف على القيثارة يَتَضَمَّنُ معرفةً حول كيف تنفيذ العزف.

معرفة الاطلاع

-- معرفة حول ما نَعْرِفُ. ومثال على ذلك: - إنسان ألي لديه خطة عن رحلة ما، بحيث انه يُمكن أن يقرأ الإشارات على طول الطريق لإكتشاف مكان وجوده.

لحل المشاكل في أنظمة الذكاء الاصطناعي يجب أن تُمَثَّل المعرفة وهناك كيانان للتعامل مع ذلك:

الحقائق

-- حقائق حول العالم الحقيقي وحول ما تُمَثَّل. هذا يُمكن أن يُعرف بمستوى المعرفة

تمثيل الحقائق

اي الشيء الذي نحن نُعالج. هذا يُمكن أن يُعتبر مستوى الرمز حيث أننا نَعْرِفُ التمثيل عادة من ناحية الرموز التي يُمكن أن تُعالج بالبرامج.

نحن يُمكن أن نُنظِّم هذه الكيانات في مستويين

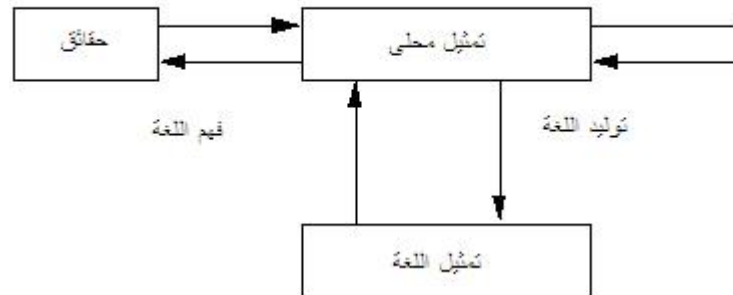
مستوى المعرفة

-- وتدل على الحقائق الموصوفة

مستوى الرمز

-- تدل على تمثيل الأجسام المُعرَّفة باستخدام الرموز التي يُمكن أن تُعالج في البرامج (انظر الرسم . 1)

برامج الإنتاج



الرسم 5 كيانان في تمثيل المعرفة

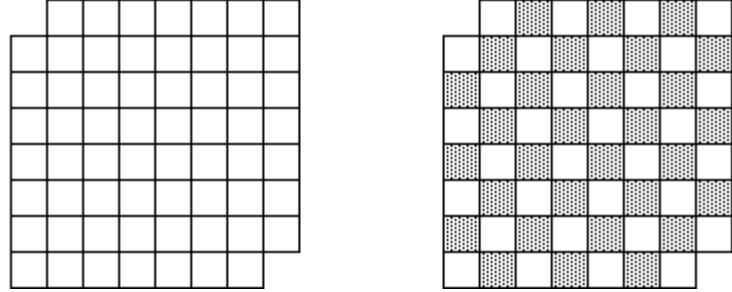
اللغة العربية أو اي لغة حية اخرى هي نوع من انواع التمثيل ومعالجة الحقائق. المنطق يُمكننا من اعتبار الحقيقة التالية: لنفترض ان حصانا اسمه ابجر، حصان(ابجر)، بما ان كل الاحصنة لها ذيل مع:

$$\forall x : \text{horse}(x) \rightarrow \text{hasatail}(x)$$

عليه مُمكن القول ان: ان ابجر له ذيل، له ذيل (ابجر).

بلستعمال خصائص اللغة امكن لجملة ك، ابجر له ذيل، ان تولد. العلاقة باللغة بين الوصف والموصوف ليست دائماً واحد لواحد لكن يمكن ان تكون العديد للعديد. كما في الجمل التالية: كل الخيول لها ذيول، وكل حصان له ذيل، كلاهما يخبر بأن كل حصان له ذيل، لكن الأولى يُمكن أن تخبر بأن كل حصان له أكثر من ذيل واحد، لنحاول الان ان نستبدل مثال الذيل بالأسنان. عندما نتعالج برامج أنظمة الذكاء الاصطناعي التمثيل الداخلي للحقائق هذه الاعتراضات الجديدة يجب أيضاً أن تكون قابلة للتفسير كاعتراضات جديدة من الحقائق.

لنأخذ مثال المشكلة الكلاسيكية للوحة الشطرنج المشوهة. حيث تم ازاله مربعات الزاوي المعاكسة. إن المهمة المُعطاة أن نغطي كل المربعات الباقية على اللوحة باحجار الدومينو بحيث ان كل حجر يغطي مربعين. مع العلم ان نُدخل الاحجار غير مسموح به. لنعتبر ثلاثة تراكيب للبيانات



الرسم 6 لوحة الشطرنج المشوهة

الأول والثاني مُصوّران في الرسم اعلاه وتركيب البيانات الثالث هو عدد المربعات السوداء وعدد المربعات البيضاء. يُفقد التخطيط الأول لون المربعات وهذا لا يؤدي لحل؛ نلاحظ وجود الألوان في التخطيط الثاني لكن ذلك لن يؤدي الى حل حيث ان عدد مربعات اللون الاسود 32 وعدد الأبيض 30 والمفروض حسب المهمة المعطاة ان تغطي كل دومنة مربع أبيض واحد ومربع أسود واحد، بمعنى اخر ان عدد المربعات يجب أن يكون متساوي لهُودي الى حل إيجابي.

استعمال المعرفة

ذكرنا سابقا وبشكل سريع اين يمكن استعمال المعرفة في أنظمة الذكاء الاصطناعي. المعرفة يمكن ان تُستعمل في المهام التالية :

التعلم

-- إكتساب المعرفة. هذا أكثر من فقط اضافة حقائق جديدة إلى قاعدة بيانات المعرفة. البيانات الجديدة يجب أن تُصنّف قبل التحزني للاسترجاع السهل لاحقا. التفاعل والاستدلال بإيجاد الحقائق لتقادي ال تكرار في قاعدة بيانات المعرفة وأيضاً لتحديث الحقائق في قاعدة البيانات.

الاسترجاع

-- مخطط التمثيل المستعمل يُمكن أن يكون له تأثير حرج على كفاءة الطريقة. بينما نجد ان البشر جيدين فيه. الكثير من أنظمة الذكاء الاصطناعي حاولت مشابهة (تقليد) الإنسان بطرق استرجاع المعلومات.

التفكيك والاستنتاج

-- استنتج الحقائق من البيانات.
لفترض ان نظام ما يعرف الحقائق التالية فقط:

- "فريد موسيقي جاز."
 - "كل موسيقيو الجاز يمكن أن يعزفوا بشكل جيد على آلاتهم."
- إذا استفسارات مثل هل فريد موسيقي جاز؟ أو هل بإمكان موسيقيو الجاز العزف على آلاتهم بشكل جيد؟ اسئلة ممكن ان نحصل على جوابها بسهولة من التراكييب وإجراءات البيانات.
- على أية حال استفسار مثل: هل بإمكان فريد ان يعزف على آله بشكل جيد؟ يتطلب التفكيك والاستنتاج.
- وهكذا يبدو واضحاً ان كل ماورد أعلاه مرتبط ببعضه البعض. على سبيل المثال، من الواضح جداً ان التعلم والتفكيك يتضمنان إسترجاعاً الخ.

خصائص أنظمة تمثيل المعرفة

نظم تمثيل المعرفة يجب ان تمتلك الخصائص التالية:

الكفاية التمثيلية

-- القدرة على تمثيل المعرفة المطلوبة؛

الكفاية الإستنتاجية

- القدرة على معالجة المعرفة الممتلئة وذلك بهدف استنتاج معرفة جديدة والتي تقابل تلك المُستنتجة من المعرفة الأصلية؛

الكفاءة الإستنتاجية

- القدرة على توجيه الآليات الإستنتاجية إلى الإتجاهات الأكثر إنتاجاً بتقزني الأدلاء الملائم؛

الكفاءة المكتسبة

- القدرة على إكتساب المعرفة الجديدة باستخدام طرق آلية محتملة لاستبدال الإعتماد على التدخل الإنساني.
حتى الآن لا يوجد نظام وحيد له كل الخصائص الواردة أعلاه.

طرق تمثيل المعرفة

لنرى بعض مخططات التمثيل سريعاً. حيث سنقوم بدراسة بعضها بتفصيل اكبر في الفصول الأخرى.

المعرفة العلائقية (المرتبطة) البسيطة

إن الطريق الأسهل لتخزين الحقائق أن يستعمل طريقة علائقية بحيث أن كل حقيقة حول مجموعة أجسام يتم عرضها بشكل منظم في الأعمدة. يعطي هذا التمثيل فرصة صغيرة للإستدلال، لكنه يمكن أن يستعمل كقاعدة المعرفة لمحركات الإستدلال. وعليه يمكن تلخيص خصائص هذا النوع من التمثيل بالتالي:

- "طريقة بسيطة لتخزين الحقائق."
- "كل حقيقة حول مجموعة الأجسام تُعرض بشكل منظم في الأعمدة (رسم 3)."
- "فرصة ضئيلة للإستدلال."
- "قاعدة معرفة لمحركات الإستدلال."

اسم الموسيقي	نوع الموسيقى	الآلة	العمر
سعيد	جاز	ترومبيت	ميت
حسام	شرقي	عود	35
تحسين	روك	جيتار	ميت
موفق	كلاسيك	ساكسافون	47

جدول 1: المعرفة العلائقية البسيطة

نحن يمكن أن نطلب استفسارات مثل:

"من ميت؟"

"الذي يلعب الجاز / بوق الخ.؟"

هذا النوع من التمثيل شعبي في أنظمة قاعدة البيانات.

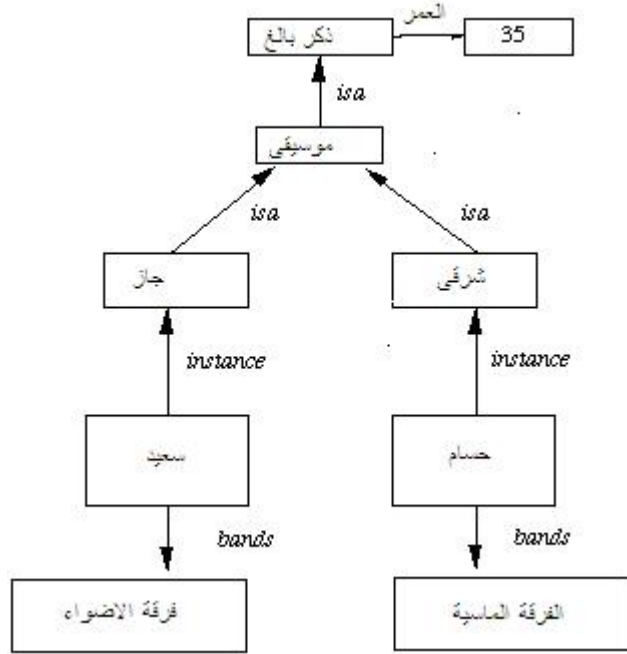
المعرفة الموروثة

المعرفة العلائقية تُعطي الأشياء:

- "خواص"
- "قيم مرتبطة مطابقة."
- تُمدد القاعدة أكثر بالسماح لآليات الاستدلال:

"ميراث الملكية"

- عناصر تَرثُ القِيَمَ بمجرد كونها أعضاء صنف.
- بيانات يجب أن تُنظَمَ في هرمية الأصناف (رسم. 4).



الرسم. 7 تدرج ميراث الملكية

- "عقد مُؤطرة -- أجسام وقيم خواص الأجسام."
- "قيم يمكن أن تكون أجسام بالخواص."
- "أسهم -- نقطة من الجسم إلى قيمتها."
- " التركيب المعروف باللقب والحشوة يُنظَم، شبكة دراسة معاني الكلمات أو مجموعة الإطار."

الخوارزمية الخاصة بليسترجاع قيمة لخاصية جسم ما:

1. جذ الجسم في قاعدة المعرفة.
2. اعط نتيجة إذا كان هناك قيمة ما.
3. فيها عدا ذلك نبحث عن قيمة الحالة وإذا لم نصل لشيء يعني فشل المهمة.
4. بخلاف ذلك رتبه إلى تلك العقدة ونبحث عن قيمة للخاصية وبعد ذلك نبلغ عنها.
5. ما عدا ذلك نتابع البحث حتى الحصول على قيمة للخاصية.

المعرفة الاستنتاجية

تمثيل المعرفة الاستنتاجية تعني تمثيل المعرفة كمنطق شكلي: لنعود الى المثال القديم

كُلّ الخيول لها ذبول: $\forall x: horse(x) \rightarrow hasatail(x)$ يؤدي الى:

- " مجموعة القواعد الصارمة.
 - يمكن أن يُستعمل لإشتقاق حقائق أكثر.
 - حقائق البيانات الجديدة يُمكن أن تُحقّق.
 - الوصول للصواب مضمون.
- "توفر العديد من إجراءات الاستدلال المؤدية إلى قواعد الأداة القياسية للمنطق.
- "مميز في انظمة الذكاء الاصطناعي. ومثال على ذلك: - مكنة إثبات النظرية.

المعرفة الإجرائية

الفكرة الأساسية:

معرفة تم تمييزها في بعض الإجراءات:

- برامج صغيرة تُعرف كيف تعمل أشياء معينة.
- ومثال على ذلك: - مُفسر لغوي في لغة طبيعية يفهم ويستنتج لاي المعرفة التي هي عبارة إسمية قد تحتوي الصفات والأسماء. حيث انه مُمثل بالدعاءات إلى الروتينات التي تُعرف كيف تُعالج المقالات والصفات والأسماء.

الفوائد من تمثيل المعرفة الاجرائية:

- "يمكن تمثيل كل من المعرفة الإرشاديّة أو معرفة مجال معين.
- "سهولة الاستدلالات المنطقية، مثل التفكيك والاستنتاج.

الأضرار:

- "الكمال -- ليس كُـلّ الحالات قليلة للتمثيل.
- "الإتساق -- ليس كُـلّ الاستنتاجات قد تكون صحيحة.
- ومثال على ذلك: - إذا كنا نعرف بأن ياغو طير هذا يسمح لنا باستنتاج بأن ياغو يُمكن أن يطير. لاحقاً نحن قد نكتشف بأن ياغو ببعاء.
- "انعدام المقياس والمعيار. بعض التغييرات في قاعدة المعرفة ربّما يكون لها تأثيرات بعيدة المدى.
- "معلومات مراقبة مُتعبة.

قضية في تمثيل المعرفة

رُجّ اسفلا بعض القضايا التي يجب أن توضح عند استعمال تقنية تمثيل المعرفة:

الخواص المهمة

-- تهتم فيما اذا كان هناك أي خواص تظهر في العديد من الأنواع المختلفة للمشكلة.

العلاقات

-- تبحث في العلاقة بين خواص جسم، مثل، التناسب العكسي، الوجود، تقنيات للتفكيك حول القيم والخواص المقيمة الوحيدة. نحن يُمكن أن نعتبر مثال نسب عكسية في فرقة (حسام، الفرقة الماسية)

$band(Hosam, Masyah\ band)$

هذا يُمكن أن يفهم على ان حسام يعزف في الفرقة الماسية أو ان فرقة حسام هي الفرقة المسماة بالفرقة الماسية. التمثيل الآخر: الفرقة = الفرقة الماسية

$band = Masyah\ band$

أعضاء الفرقة = حسام ، فريد، شريف... الخ .

$band-members = Hosam, Fared, Shareef, \dots$

خاصية الجزيئات

--للدلالة على المستوى الذي يجبُ تمثيل المعرفة به ماهي البدايات . إختيَار الجزيئات البدائية لتمثيل هي مفاهيم أساسية مثل التوقف، الرؤية، التشغيل. كمثال على ذلك نأخذ اللغة الإنجليزية هي لغة غنية جداً بأكثر من نصف مليون كلمة نحن سنجد صعوبة في التعبير، أي الكلمات تعتبر كلمات للإختيَار البدائي في سلسلة من الحالات. كأمثلة يمكن عرض التالي:
إذا سامي يطعم كلب، يُمكن أن يُصبح:

feeds(Sami, dog)

أطعم (سامي، كلب)
إذا يعطي سامي الكلب عظم:
gives(sami, dog, bone)
يعطي (سامي، كلب، عظم) هل هذه الجملة تؤدي نفس معنى الجملة الأولى؟
هل يعني إعطاء الطعام الإطعام؟
إذا $give(x, food) \rightarrow feed(x)$ بمعنى اذا كان اعطاء س الطعام يؤدي الى اطعام س هذا يدل على اننا نتقدم.
لكننا نحتاج لإضافة بعض القواعد الإستنتاجية.
لنأخذ مثال اخر كيف نمثل علاقة شخص بأخر، هل زيد ابن عم عمر كيف نمثل هذا؟
 $Zayd = son(brother \text{ or } sister(father \text{ or } mother(Amr)))$
زيد = ابن (أخ أو أخت (أب أو أم (عمر))).
من الواضح ان المستويات المنفصلة لفهم مستويات مختلفة من الجزيئات الابتدائية تحتاج الى العديد من القواعد لربط البدايات المماثلة سوية.
من الواضح ان هناك مشكلة نحزني محتملة والسؤال يجب أن يكون ماهو مستوى الفهم المطلوب.

الخلاصة

في هذا الفصل تأكدنا من:

- الحاجة الى المعرفة في برامج التفكير والاستنتاج.
- العديد من القضايا التي من الضروري أن نعتبر عند تقرير مخطط ما لتمثيل المعرفة.
- قدمنا مفهوم بيانات الحشوة والشق في المعرفة الموروثة. في الفصلين القادمين سندرس أنواع مختلفة من تمثيل الحشوة والشق يبدان بالشبكات والإطارات والانتقال الى التمثيل المبني على التبعية التصويرية الأقوى.

تمثيل المعرفة المنطقي

ذكرنا بشكل سريع كيف ان المنطق يمكن أن يستعمل لتمثيل الحقائق البسيطة. في هذا الفصل نحن سنبرز مبادئ رئيسية إشتكت في تمثيل المعرفة في الواقع المسندات المنطقية سيجتمع في مخططات تمثيل المعرفة الأخرى وطرق التفكير والاستنتاج.
سنعرض هنا رموز المنطق القياسية التالية التي استعملت في هذا الفصل:

مهما يكن

∇

على الأقل

∃

يؤدي

→

نفي

¬

أو

∨

و

∧

لنأخذ الآن مثال عن كيفية استعمال المسندات المنطقية لتمثيل المعرفة. هناك طرق أخرى لكن هذا الشكل معروف.

منطق مسند

مثال

لنعتبر التالي:

- "أمير - نجم كبير.
 - "النجوم الكبار أغنياء.
 - "النفاس الاغنياء عندهم سيارات سريعة.
 - "تستهلك السيارات السريعة الكثير من البنزين.
- ولنحاول الإستنتاج: تستهلك سيارة الأمير الكثير من البنزين.
لذا نحن يمكن أن نترجم أمير نجم كبير إلى: $mega_star(prince)$ والنجوم الكبار أغنياء إلى :

$$\forall m: mega_star(m) \rightarrow rich(m)$$

والنفاس الاغنياء عندهم سيارات سريعة، البديهية الثالثة أكثر صعوبة:

- " هل السيارات هي علاقة ولذا $car(c,m)$ تخبر بأن الحالة c هي سيارة m. أو
 - "هل السيارات دالة؟ لذا نحن عندها $car_of(m)$.
- لنفترض ان السيارات علاقة فأن البديهية 3 قد تُكتب:

$$\forall c, m: car(c, m) \wedge rich(m) \rightarrow fast(c).$$

إن البديهية الرابعة هي بيان عام حول السيارات السريعة. لنفترض ان $consume(c)$ متوسط استهلاك تلك السيارة للبنزين. ثم نحن يمكن ان نكتب :

$$\forall c: [fast(c) \wedge \exists m: car(c, m) \rightarrow consume(c)]$$

لنسأل الآن هل الأمير لديه سيارة؟ نحتاج الى الدالة car_of بعد ذلك :

$\forall c: car(car_of(m), m) .$

إن المجموعة النهائية من المسندات :

$mega_star(prince) \forall m: mega_star(m) \rightarrow rich(m) \forall c: car(car_of(m), m) . \forall c, m: car(c, m) \wedge$
 $rich(m) \rightarrow fast(c) . \forall c: fast(c) \wedge \exists m: car(c, m) \rightarrow consume(c)$

باعطاء ذلك يمكن ان نصل للنتيجة التالية :

$consume(car_of(prince)) .$

علاقات الحالة والمضمون

تُلعبُ خاصيتي علاقات الحالة والمضمون دور مهم في العديد من سمات تمثيل المعرفة. إن السبب وراء هذا الدور المهم هو شمولهم على ميزات ميراث الملكية. المضمون:

--يُستعمل لبيان مضمون الصنف، ومثال على ذلك : $isa(mega_star, rich)$.
 علاقة الحالة :

--يُستعمل لبيان عضوية الصنف، ومثال على ذلك : $instance(prince, mega_star)$.

من الوارد أعلاه يتبين انه من السهل معرفة كيف يُمثّل هذا في المنطق المسند.

التطبيقات والإمتدادات

- "يُمدّد الطلب الاول المنطقي مبدائيا الى المسندات للسّماح ب:
 ○ وظائف – تعطي اجسام كنتائج وليس فقط صح / خطأ.
 ○ يساوي مسند إضافي.
- "حلّ مشكلة وإثبات نظرية -- مناطق تطبيق كبيرة.
- " تعرية نظام التخطيط الآلي يوظف الطلب الاول المنطقي لتحسين تحليل نهايات وسائل تخطيط (GPS).
 هذا الدمج زوّد ببحث إرشادي قوي جداً.
- "أنظمة إجابة على أسئلة واستفسارات.

تمثيل المعرفة التصريحي و الإجرائي

تمثيل المعرفة التصريحي:

- "التمثيل الساكن -- معرفة حول الأجسام، والأحداث الخ. وعلاقاتهم والحالة المعطاة.
- "يُتطلّب برنامج لمعرفة ما العمل بالمعرفة وكيف يعمل ه ذا.

التمثيل الإجرائي:

- معلومات سيطرة وتحكم ضرورية لإستعمال المعلومات المُضمّنة في المعرفة نفسها. ومثال على ذلك: -
 كيف يجذ الحقائق ذات الصلة، والإستدلالات الخ.
- يُتطلّب مفسراً لتنفيذ التعليمات المحددة في المعرفة.

مثال:

لمعرفة احتياجات فارز ألفبائي نحتاج:

- معرفة ضمنية بأن "أ" تأتي قبل "ب" الخ.
- هذا سهل -- بمقارنة عدد صحيح من رموز (أسكي) له أ، ب الخ.
 ○ كل البرامج تحتوي على معرفة إجرائية من هذا النوع.
- المعلومات الإجرائية هنا هي تلك المعرفة حول كيف تُرتّب حسب الحروف الأبجدية تظهر واضحة في إجراء الألفبائية.
- أي نظام تصريحي يجب يعتمد على حقائق واضحة مثل "أ" يأتي قبل "ب"، "ب" قبل "ث" الخ. .

تمثيل كيف تستعمل المعرفة

نحتاج هنا لتمثيل كيفية السيطرة على مسار المعالجة:

الإتجاه

-- يُشير إلى إتجاه النتيجة المضمنة والتي يُمكن أن تُستعمل. ومثال على ذلك: - لإثبات ان الشيء يُمكن أن يُطير يكفي عرضه لطير $bird(x) \rightarrow fly(x)$.

المعرفة للوصول الى الهدف

-- يُحدّد ماهي المعرفة المطلوبة للوصول الى هدف معيّن. على سبيل المثال لإثبات ان الشيء هو طير يجب ان نستخدم الحقائق: له - أجنحة و له - ريش.

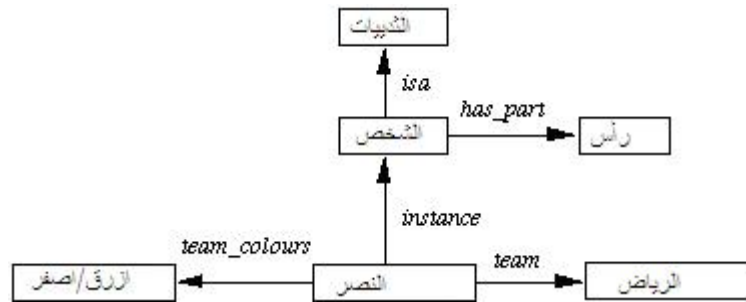
بنية الشق الضعيف والحشوة

لماذا نستخدم تركيب البيانات هذا؟

- نتمكن هذه البنية من استرجاع قيم الخاصية بسرعة، ومنها:
 - المزاعم المُفهرسة بالكيانات
 - المسندات الثنائية المُفهرسة بالحجة الأولى. ومثال على ذلك: - فريق (النصر، الرياض).
 - خصائص العلاقات سهلة للوصف.
 - يسمح بسهولة التعويض حيث انه يتخذ سمات البرمجة الموجهة للأشياء.
 - عليه ممكن قول التالي فيما يخص نظرية الشق الضعيف والحشوة:
 - يعتبر الشق قيمة بخاصية ازدواجية في شكله الأسهل.
 - تعتبر الحشوة القيمة التي يُمكن أن يأخذها الشق -- يُمكن أن تكون قيمة عدديّة، نصية (أو أي نوع بيانات أخرى) أو مؤشر إلى شق آخر.
 - تركيب الشق الضعيف والحشوة لا يعتبر محتوى التمثيل.
 - نحن سندرس نوعان من هذه البنية والتركيب:
 - شبكات المعاني.
 - الإطار.
- شبكات المعاني
- إنّ الفكرة الرئيسية لشبكات المعاني هي:
- معنى فكرة أو تصور ما، يأتي من علاقته بالمفاهيم والتصورات الأخرى.
 - المعلومات مخزنة برابط العقد مع أقواس مصنفة.

التمثيل في شبكة المعاني

ان الخواص الفيزيائية لشخص ما يُمكن أن نتمثل كما في الرسم 8.



الرسم 8 شبكة سيمانطيقية (لفظية)
هذه القيم يُمكن أيضاً أن تُتمثل في المنطق الثلاثي:

isa(person, mammal), instance(Al-Nasr, person) team(Al-Nasr, Riyadh)

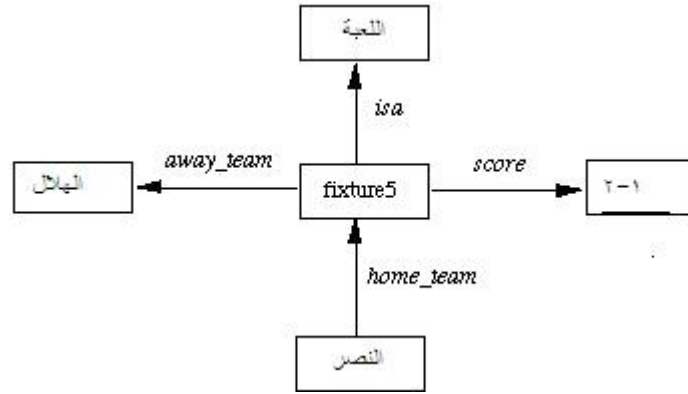
المضمون (شخص، الثدييات)، الحالة (النصر، شخص) فريق (النصر، الرياض) بمعنى اخر يمكن القول ان الشخص هو من الثدييات له رأس لاعب في فريق النصر، الذي لون ملابسه ازرق/اصفر – الرياض.

رأينا كيف ان مسند تقليدي مثل *lecturer(Jad)* يُمكن أن يُكَنَّب كحالة (*Jad, lecturer*) ، إستدعاء المضمون (*isa*) والحالة، يمثل الصفة الوراثية ويعتبر مستعمل بشكل كبير في العديد من مخططات تمثيل المعرفة. لكن لدينا مشكلة هنا : وتكمن بكيفية أخذ أكثر من مكان لمسند ما في الشبكات السيمانطيقية؟ ومثال على ذلك:

score(Al-Naser,Al-Helal, 2-1)

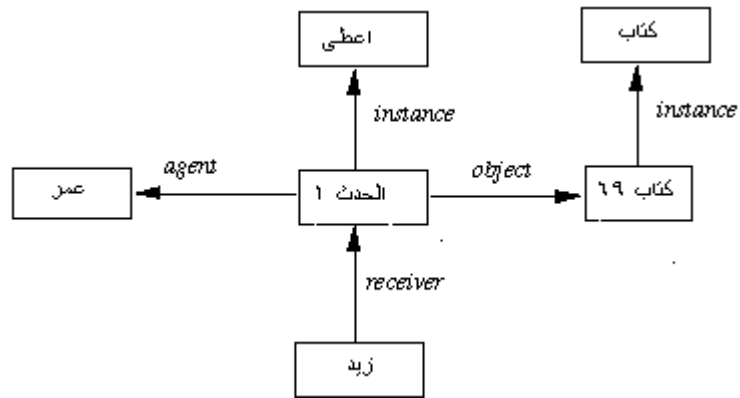
تحل هذه المشكلة حسب التالي:

- بإنشاء عُقد جديدة لتمثيل الأجسام الجديدة التي تحتوي أو تشير إلى المعرفة، game و fixture في المثال التالي.
- بالربط بين المعلومات والعقد وملاً الشقوق (الرسم: 9).



الرسم. 9 شبكة سيمانطيقية لمسند بـ n مكان

بينما بمثال أكثر تعقيداً كما في الجملة: أعطى عمر زيدا كتاباً هنا لدينا عدة سمات للحدث.



الرسم. 10 شبكة سيمانطيقية لجملة ما

الإستدلال في شبكة المعاني

آلية الإستدلال الأساسية هي : اتباع الوابط بين العقد.

هناك مبدآن لتنفيذ ذلك:

البحث التقاطعي

-- الفكرة التي يقوم عليها البحث التقاطعي تتمثل بتنشيط عُقدتين وتَديدِ عِلَاقَاتٍ تقاطعِهم بايجاد علاقتهم مع الأشياء. يتم أنخَازُ هذا بتخصيص بطاقة خاصة لكل عقدة تم المرور عليها.

العديد من الفوائد تَضمَرُ في البنية المنظمة على أساس الكيان وفي البنى ذات خاصية التطبيق المتوازي السريع. على أية حال الاستفسارات ذات البنى المنظمة جدا بحاجة الى شبكات منظمة إلى حد كبير.

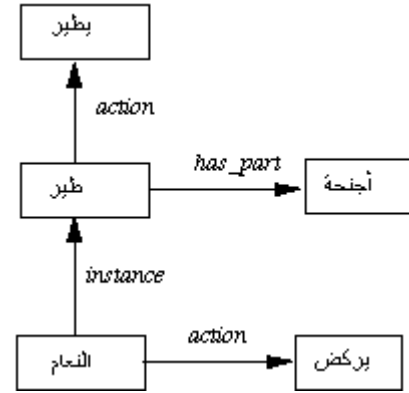
الميراث

-- تمثيل المضمون والحالة يُزودان ببلية لتطبيق نظرية الميراث.

يُزود الميراثُ أيضا بوسائل التَعاُمَل مع التَفَاقِي والاستنتاج المهمل. ومثال على ذلك: - نحن يُمكن أن نُمثَل التالي في شبكة المعاني:

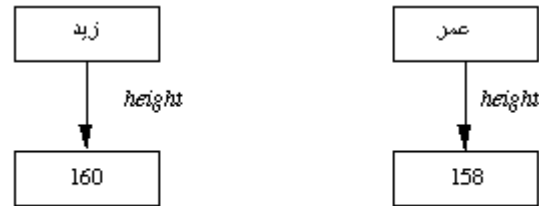
- النعام طير.
- بشكل نموذجي الطيور تطير ولها أجنحة.
- النعام يركض.

في الشبكة السيمانطيقية التالية:



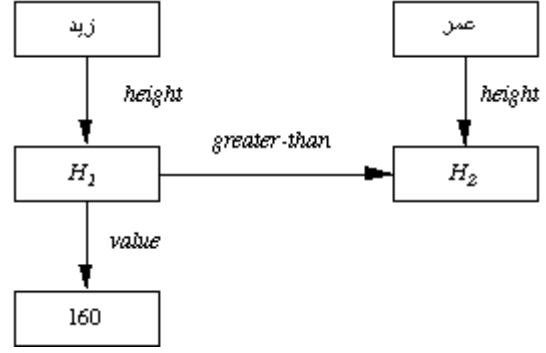
الرسم . 11 شبكة سيمانطيقية لتفاني مهمل

لعمل إستدلالات اكيدة نحن سَنَحتاج للتمييز بين الوابط الذي يُعرِّف كيان جديد ويَحمِلُ قيمته والنوع الآخر للوالب الذي يَتَعلَق بالكيانين الحاليين. لنأخذ المثال المعروض حيث أن طول شخصين حُدد وعلينا أن نُقارَنَهم. نَحتاج عَقْدَ إضافية للمفهوم بالإضافة إلى قيمتها.



الوسم 12. طول شخصان

الإجراءات الخاصة تَحتاج لمعالَجة هذه العَقْد، لكن بدون هذا الإمتياز، التحليل سَيَكون محدود جداً.



الرسم 13. المقارنة بين طولين
تم الرمز الى الطول الاول H1 والثاني H2 وسيتم المقارنة بينهم "بأطول من"

تمديد الشبكات السيمانطيقية

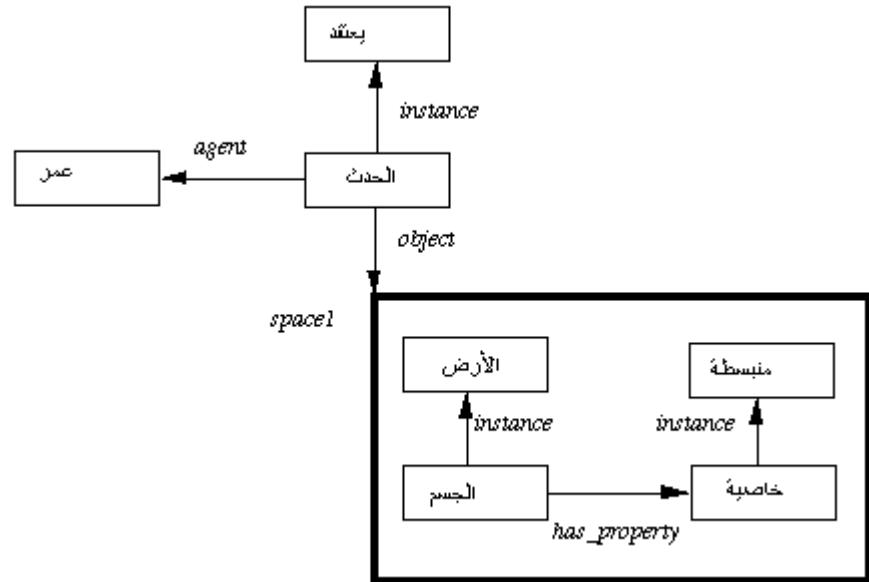
سنرى هنا بعض الإمتدادات الخاصة بدراسة شبكات المعاني والتي تتغلب على بضعة مشاكل أو توسع تعبيرهم من المعرفة.

الشبكات المُقسَّمة: شبكات المعاني المقسمة تسمح بالتالي:

- المقترحات التي سنُضِلُّ بدون إلزام إلى الحقيقة.
- التعابير التي سنُحدِّد.

الفكرة الأساسية: تتمثل بتجزئة الشبكة إلى المساحات التي تشمل مجموعات العُقَد والأقواس ويُعتبر كل حيز كعقدة.

لنعتبر التالي: يُعتقد عمر بأن الأرض مستوية. نحن يُمكن أن نرمز الى هذا المقترح "الأرض مستوية (منبسطة)" في مساحة ويكون لها العُقَد والأقواس التي نُقل هذه الحقيقة (رسم 14). نحن يُمكن أن يكون عُنْدنا العُقَد والأقواس المطلوبة لربط هذا الفضاء ببقية الشبكة لتمثيل إعتقاد عمر.



الوسم 14 شبكة مُقسَّمة

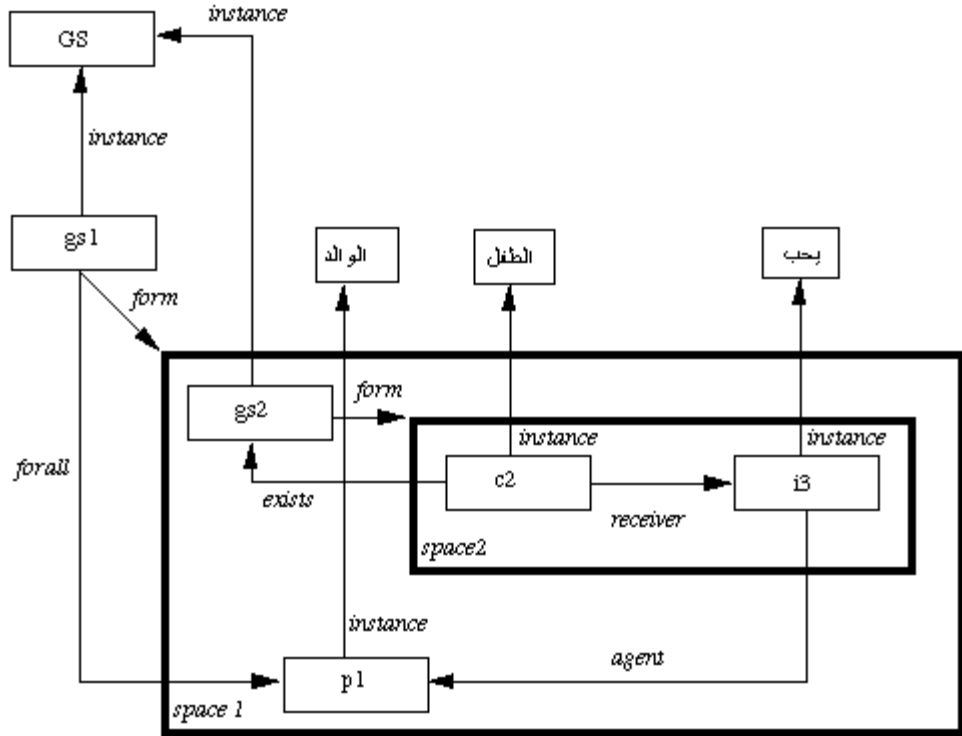
لنعتبر الان التعبير المُحدَّد التالي : يَحِبُّ كُلُّ والدين طفلهم. لتمثيل هذا التعبير نحن نعمل التالي:

- ننشأ البيان العام، GS، بصنف خاص.
- نعتبر g عقدة حالة من GS.
- كُل عنصر سيكون عِنْدَهُ على الأقل خاصيتين:
o الشكل الذي يحدد العلاقة الموكدة.
o واحد أو أكثر من ارتباطات (مهما يكن \forall) أو (على الأقل \exists) -- هذا يُمثِّل متغيرات قابلة للقياس في مثل هذه البيانات ومثال على ذلك:

$$x, y \text{ in } \forall x \text{ parent}(x) \rightarrow \exists y : \text{child}(y) \wedge \text{loves}(x, y)$$

هنا نحن يَحِبُّ أَنْ نَبْنِي فراغان واحد لكل من x, y الوالدان والطفل. نلاحظ هنا : انه يُمكن أَنْ نَبْنِي المتغيرات كمتغيرات مشروطة بشكل وجودي ونَبْنِي حالة الحب التي سيكون عِنْد المصدر p والمستلم b لكل أصل p الذي يُمكن أَنْ يُبسِّط الشبكة.

أيضاً إذا غيّرنا الجملة إلى : كُل والد يَحِبُّ طفلاً فإن العقدة هنا و التي تُتصرّف وفقها (الطفل) تقع خارج شكل البيان العام. حيث لم يُنظر إليها كمتغيرات مشروطة بشكل وجودي والتي قيمتها قد تعتمد على المصدر الذي هو الوالد هنا. فمن الاكيد ان الوالد يحب طفله، لكنه ليس بالاكيد انه يحب الاطفال الاخرين بشكل عام. لذا نحن يُمكن أَنْ نَبْنِي شبكة مقسمة كما في الرسم. 15



الرسم. 15 شبكة مُقسَّمة

الإطر

الإطر يُمكن أيضاً أَنْ تُعتبر امتداداً للشبكات السيمانتيقية. في الحقيقة الإمتياز بين شبكة المعاني والإطر ليس واضحاً أين ينتهي. مبدائياً نحن نستخدم شبكة المعاني لُمثِّل الإرتباطات بين الأجسام المحددة. لعلها أصبحت المهام أكثر تعقيداً

فأن من الضروري أن يكون التمثيل أكثر تنظيماً. في النظم الأكثر تنظيماً يصبح استعمال الاطر أكثر فائدة. عليه يمكن تعريف الاطر بأنها مجموعة الخواص أو الشقوق والقيم المرتبطة التي تصف بعض الكيانات من العالم الواقعي. الاطر بشكلها المفرد لاتعتبر ذات فائدة كبيرة، لكن الأنظمة المبنية على اساس الاطر تعتبر طريقة قويّة لتفسير (ترميز) المعلومات لدعم التفكير والاستنتاج. نظرية المجموعات تعتبر قاعدة جيدة لفهم أنظمة الاطر. كل إطار يُمثّل:

- صنف (مجموعة)، أو
- حالة (عنصر من صنف).

تمثيل المعرفة بالاطر

لنأخذ المثال التالي:

Person

<i>isa:</i>	<i>Mammal</i>
-------------	---------------

<i>Cardinality:</i>	<i>...</i>
---------------------	------------

Adult-Male

<i>isa:</i>	<i>Person</i>
-------------	---------------

<i>Cardinality:</i>	<i>...</i>
---------------------	------------

Football-Player

<i>isa:</i>	<i>Adult-Male</i>
-------------	-------------------

<i>Cardinality:</i>	<i>...</i>
---------------------	------------

Height:

Weight:

Position:

Team:

Team-Colours:

Forwards

<i>isa:</i>	<i>Football-Player</i>
-------------	------------------------

<i>Cardinality:</i>	<i>...</i>
---------------------	------------

Tries:

Al-Nasr

instance: Forwards
Height: 6-0
Position: Centre
Team: Riyadh-SAT
Team-Colours: Yellow/Blue

Football-Team

isa: Team
Cardinality: ...
Team-size: 25
Coach:

Al-Naser

Instance:

Team-size

Coach:

Players:

Fottball-Team

25

Artur Jorge

{Moh. Khojali, Ahmad Al-Ajami, Faisal Saif.....}

يعتبر هذا المثال نظام إطر بسيط
نلاحظ هنا ان الاطر : *Person* (شخص) ، *Adult-Male* (ذكر بالغ) ، *Football-Player* (لاعب كرة قدم) و
Football-Team (فريق كرة قدم) كلها أصناف. والإطر : سعد الحارثي (لاعب في فريق النصر) - *Saad Al-Harhi*
و الفرق السعودية *Riyadh-SAT* هي حالات.
نلاحظ ايضا :

- علاقة المضمون isa في الحقيقة هي علاقة المجموعة الثانوية.
- علاقة الحالة *instance* في الحقيقة هي عنصر.
- خاصية المضمون isa تمتلك خصائص انتقالية. هذا يدل عليه: روبرت ظهري والظهير هو لاعب كرة قدم والذي تباعاً هو ذكر بالغ وأيضاً شخص.

Saad Al-Harhi Forwards , Forwards is a Football-Player, Adult-Male, Person.

بمعنى ان سعد الحارثي لاعب هجوم في فريق كرة القدم وهو شخص ذكر بالغ.

- كلنا علاقتي المضمون isa والحالة لهما العكوس التي تدعو الفئات الفرعية أو كل الحالات.
- هناك الخواص التي ترتبط بالصنف أو المجموعة وتعتبر اساسية وجوهرية ومن الناحية الأخرى هناك الخواص التي يملكها كل عضو في الصنف أو المجموعة.

الاختلافات بين المجموعات والحالات

من المهم بأن يفهم هذا الاختلاف بشكل واضح.
Riyadh-SAT يُمكن أن تعتبر مجموعة اللاعبين أو كحالة فريق كرة قدم.

إذا *Riyadh-SAT* هي صنف فإن

- اللاعبين سيكونون حالات الصنف
- هذا الصنف لا يُمكن أن يكون فئة فرعية من فريق الكرة قدم ما عدا ذلك عناصره ستكوّن أعضاء فريق الكرة قدم أو هذا ما لا نريده.
- بدلاً من ذلك نجعله فئة فرعية من لاعب كرة قدم وهذا يسمح للاعبين ورث الخصائص الصحيحة التي نتكئنا من جعل النصر يورث المعلومات حول الفرق.
- هذا يعني بأن النصر حالة من حالات فريق كرة قدم.
- لكن هنا لدينا مشكلة:

- أي صنف هو مجموعة وعناصره لئيم خصائص.
 - نحن نرغب بلستعمال الميراث لمنح القيم على أعضائه.
 - هناك بعض الخصائص التي تملكها المجموعة أو الصنف، مثل مدير الفريق.
- لهذا السبب نحن نريد جعل فرق كرة القدم بالرياض كمجموعة ثانوي من لاعبي صنف واحد وحالة من حالات الفريق. الحل لهذه المشكلة يكمن في : الصنف المركب والذي يعرف بالصنف الخاص والذي يعتبر كل واحد من عناصره صنفاً بحد ذاته.
- لنعتبر الآن فرق كرة القدم ك:

Class

Instance: **Class**

Isa: **Class**

Cardianlity:

Team:

Instance: **Class**

Isa: **Class**

Cardianlity: { The number of teams}

Team-Size 25

Football- Team

Isa: **Team**

Cardianlity: { The number of teams}

Team-Size: 25

Coach:

Al-Naser

Instance: **Football-Team**

Team-Size: 25

Coach: **Artur Jorge**

Ahmad Al-Ajami

Instance: **Defender**

Height: 170

Position: **Defender**

Team: **Al-Naser**

Team-Colors: **Blue/ Yellow**

نظام إطار الصنف المركب

- الصنف المركب الأساسي هو صنف، وهذا يُسمَح لنا بـ :
- تعرّيج الأصناف التي هي حالات لأصناف أخرى.
 - وراثتها الخصائص من هذا الصنف.
- وراثتها القيم الأصلية تحدث عندما عنصر واحد أو صنف يكون حالة الصنف المورث.

شقوق كأجسام

- كَيْفَ يمكن تمثيل الخصائص التالية في الإطار؟
- خواص مثل الوزن، العمر تكون ملحقاتاً ونصبح مفهوم.
 - فرض قيود على القيم لتمثيل في حالة خاصية العمر أن تكون أقل من مائة.
 - قواعد لصالح ميراث القيم مثل الأطفال الذين يرثون أسماء الوالد
 - قواعد لحساب القيم
 - العديد من القيم للشق.
- يعتبر الشق العلاقة التي تُجدول وتنظم مجال الاصناف إلى مداها من القيم.
- تعتبر العلاقة مجموعة الأزواج المنظمة عليه فإن علاقة واحدة هي مجموعة ثانوية من أخرى.
- باعتبار ان الشق مجموعة، مجموعة كل الشقوق يُمكن تمثيلها بالصنف المركب والمسمى Slot.

لنفترض التالي:

SLOT

isa: *Class*

instance: *Class*

domain:

range:

range-constraint:

definition:

default:

to-compute:

single-valued:

Coach

instance: *SLOT*

domain: *Football-Team*
 range: *Person*
 range-constraint: λx (experience x .manager)
 default:
 single-valued: *TRUE*

Colour

instance: *SLOT*
 domain: *Physical-Object*
 range: *Colour-Set*
 single-valued: *FALSE*

Team-Colours

instance: *SLOT*
 isa: *Colour*
 domain: *team-player*
 range: *Colour-Set*
 range-constraint: *not Pink*
 single-valued: *FALSE*

Position

instance: *SLOT*
 domain: *Football-Player*
 range: { *Forwards, Forward, Reserve* }
 to-compute: λx *x.position*
 single-valued: *TRUE*

نُلاحظ التالي:

- حالات الصنف المركب *SLOT* شقوق
- الخواص التي كل حالة سترثها ترتبط بالشق *SLOT*.
- كُّل شقّ له مجال ومدى.

- الهدى يُقسَّم إلى جزئين الأول هو صنف من العناصر والآخر هو القيد الذي يعتبر تعبير منطقي في حال ان النتيجة تصل الى الصبح.
 - إذا كان هناك قيمة مبدائية فإنه يجب إعادة استخدامها طالما الحالة لم تأخذ قيمتها الخاصة.
 - خاصية الحساب تتضمَّن إجراء لحساب قيمته. ومثال على ذلك: - في الموقع حيث يتعملُ ترقيم النقطة لتخصيص القيم إلى شقٍّ ما من إطار.
 - الانتقال خلال شقوق القوائم الأخرى التي تشتق منها القيم من الميراث.
- تراكيب الشقّ القوي والحشوة**

- تراكيب الشقّ القوي والحشوة بصفة نموذجي تغطي التالي :
- "تمثيُّ الصلات بين الأجسام طبقاً للقواعد الصرامة.
 - "أفكار معينة عن أنواع الأجسام والعلاقات بينهم.
 - "تمثيُّ المعرفة حول الحالات المشتركة.

التبعية التصورية

طوّرت التبعية التصورية أصلاً لتمثيل المعرفة المكتسبة من مدخلات اللغة الطبيعية. إن أهداف هذه النظرية:

- المساعدة في رسم الاستدلال من الجمل.
 - لإستعمالها في المدخلات الأساسية بشكل مستقل عن بعض كلمات الجمل.
 - "بمعنى آخر: يمكن القول بأنه في حال وجود جملتين أو أكثر واللّتان/ الذين تحمّلان/ يحملون نفس المعنى، فإنه يجب أن يكون لها فقط تمثيل واحد من ذلك المعنى.
- هذه النظرية مستخدمة في كثير من البرامج التي تعني بفهم اللغة الإنجليزي على سبيل المثال الأنظمة التالية: (MARGIE, SAM, PAM). نظرية التبعية التصورية طوّرت من قبل Schank.
- التبعية التصورية تُروّدا:
- بالتركيب الذي يتم به حفظ المعلومات الممثلة للعقد.
 - مجموعة معينة من معلومات البدايات.
 - المستوى المحدد و المعطى من الجزئيات.
- الجمل مُمثلة كسلسلة من التخطيطات التي تُصوّر الأعمال التي تستعمل كلتا الحالتين الطبيعية المُجرّدة والحقيقية.
- الوسيلة والأجسام مُمثلة
 - الأعمال تُعرّز من مجموعة الأفعال الإبتدائية التي يُمكن أن تُعدّل بالزمن.
- أمثلة الأفعال الإبتدائية:

Atrans

-- نقل علاقة مُجرّدة. ومثال على ذلك: - يعطي.

Ptrans

-- نقل الموقع الطبيعي لجسم. ومثال على ذلك: - يذهب.

PROPEL

-- تطبيق قوة طبيعية على جسم. ومثال على ذلك: - دفع.

Mtrans

-- نقل المعلومات العقلية. ومثال على ذلك: - يُخبر.

Mbuild

-- إستنتاج وإنشاء معلومات جديدة من معلومات قديمة. ومثال على ذلك: - يُقرّر.

SPEAK

-- اصدار صوت، الكلام. ومثال على ذلك: - رأي.

ATTEND

-- تركيز الانتباه على محفّز ما. ومثال على ذلك: - الاستماع، المشاهدة.

MOVE

-- حركة عضو جسم ما من قِبل صاحب العضو. ومثال على ذلك: - لكمة، ركلة.

GRASP

-- الهمثل الذي يُدركُ الأشياء. ومثال على ذلك: - فاصل.

INGEST

-- الشخص الذي يَأْكُلُ شيئاً ومثال على ذلك: - يَأْكُلُ.

EXPEL

-- الشخص الذي يَتَخَلَّصُ مِنْ أشياء مِنْ الجسم. ومثال على ذلك: - قضاء الحاجة .

سِتَّةُ أصنافٍ تصويريةٍ إِبْدَائِيَّةٍ تُزوِّدُ كَتْلَ الهَاءِ والتي تعتبر مجموعةَ التَّبَعِيَّاتِ الجائِزةِ في المفاهيم في جملة:

PP

-- أجسام من عالم حقيقي.

ACT

-- أعمال من عالم حقيقي.

PA

-- خواص الأجسام.

AA

-- خواص الأعمال.

T

-- الوقت.

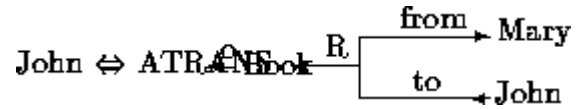
LOC

-- الموقع.

كَيْفَ رُوبِطَ هَذِهِ الْأَشْيَاءُ بِبَعْضِهَا الْبَعْضُ؟

لِنَأْخُذِ الْمَثَالَ التَّالِيَّ:

جون يَعْطِي ماري كِتَابَ



• تُشِيرُ الْأَسْهُمُ إِلَى إِتْجَاهِ التَّبَعِيَّةِ. وَالرَّمُوزُ فِي الْاَعْلَى تُشِيرُ إِلَى بَعْضِ الْعِلَاقَاتِ:

o

-- جسم.

R

-- متلقي، متسلم.

I

-- أداة ومثال على ذلك: - يَأْكُلُ بِالْمَلْعَقَةِ.

D

--الوجهة المقصودة ومثال على ذلك: -إِلْهَابِ إِلَى الْبَيْتِ.

• الْأَسْهُمُ الْمَضَاعِفَةُ (⇐) تُشِيرُ إِلَى الصَّلَاتِ الْمَزْدُوجَةِ بَيْنَ الشَّخْصِ (PP) وَالْعَمَلِ (ACT).

• الْأَعْمَالُ مَبْنِيَّةٌ مِنْ مَجْمُوعَةِ الْأَفْعَالِ الْإِبْدَائِيَّةِ. مَعَ الْعِلْمِ أَنَّ هَذَا يُمَكِّنُ أَنْ يُعَدَلَ بِالزَّمَنِ.

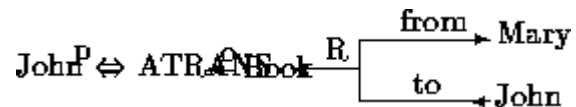
إِنَّ إِسْتِعْمَالَ الْوَقْتِ وَالْمَزَاجِ فِي وَصْفِ الْأَحْدَاثِ مَهْمَةٌ جَدًّا وَ قَدْ أورد Schank المَعْدَلَاتِ التَّالِيَةَ فِي هَذَا الْخُصُوصِ:

p

-- ماضي

f
 -- مستقبل
 t
 -- إنتقال
 t_o
 -- إنتقال بداية
 t_f
 -- إنتقال نهاية
 k
 -- إستمرار
 ؟
 -- ضمير إستفهام
 /
 -- سلبي
 delta
 -- بدون وقت، سرمدى
 c

-- شرطي
 ملاحظة : غياب المؤشر الدال على الزمن يعني اننا نقصد الزمن الحالي.
 لذا يكون المثال المعطى أعلاه في الزمن الماضي كالتالي:
 جون أعطى ماري كتاباً أصبح:



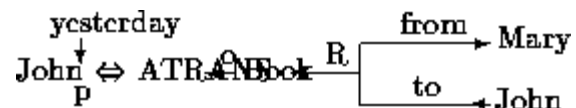
هنا نجد جسم (ACTOR)، PP وعمل، ACT، وبمعنى آخر: .PP \Leftrightarrow ACT. السهم الثلاثي (\Leftrightarrow) أيضاً هو رابط مزدوج لكن بين جسم، PP، وخاصيته، PA. وبمعنى آخر: .PA \Leftrightarrow PP.

هذا يُمثّل تبعيات نوع المضمون. ومثال على ذلك: -

Sami \Leftrightarrow lecturer سامي مُحاضر.
 الحالات الابدائية تُستعمل لوصف الكثير من الأوصاف مثل الإرتفاع، الصحة، الحالة العقلية، الحالة الفيزيائية.
 هناك المزيد من الحالات الفيزيائية الأكثر من الأعمال الابدائية. حيث يستخدم لها مقياس عددي.
 ومثال على ذلك:

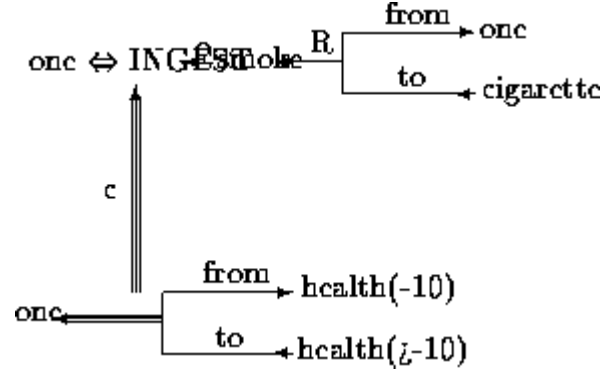
Sami \Leftrightarrow height(+10) سامي هو الأطول، حيث ان مؤشر طوله عالي جداً.
 Sami \Leftrightarrow height(< average) سامي قصير او اقصر من المعدل.
 Ali \Leftrightarrow health(-10) علي مريض، حيث ان صحته بمؤشر سلبي كبير.
 Dave \Leftrightarrow mental_state(-10) سامي متوتر ، حيث ان حالته الذهنية بمؤشر سلبي كبير.
 Vase \Leftrightarrow physical_state(-10) المزهرية مكسورة، حيث ان مؤشر حالتها الفيزيائية بقيمة سلبية كبيرة جداً.

نحن يُمكن أن تُحدّد ايضاً أشياء مثل زمن الحدث في هذا النوع من العلاقة.
 على سبيل المثال: جون أعطى ماري الكتاب أمس



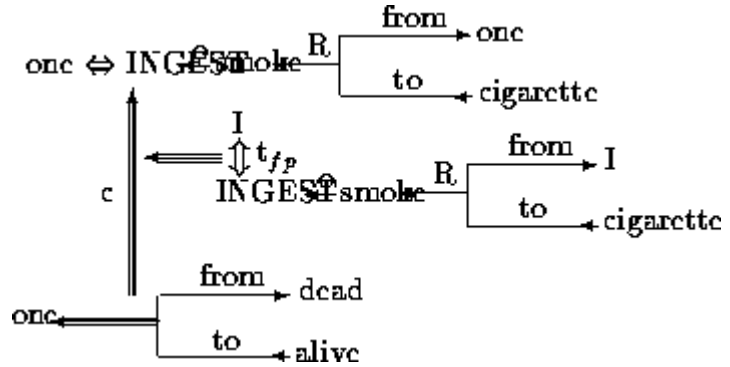
سنأخذ الآن جملة أكثر تعقيداً: انا توقفت عن التدخين، حيث انه ممكن ان يسبب الموت. لنرى الان كيف يمكن ان نمثل ذلك الإستدلال ، التدخين يسبب الموت:

- إستعمل الفكرة لتطبيق المعرفة.
- إستعمل الفعل الإبقائي INGEST لتدخين شخص ما (One) سيجارة.
- القتل او الموت هو إنتقال من حالة حي إلى حالة ميت. نستخدم أسهم ثلاثية للإشارة على الإنتقال من حالة إلى أخرى.
- لدينا هنا ايضا شرط ، c رابط سببي. يُشير السهم الثلاثي إلى التبعية من مفهوم على آخر.



لإضافة حقيقة أن الشخص قد توقف عن التدخين:

- يجب استعمال قواعد مماثلة للإشارة ضمناً إلى أن الشخص يُخّن السجائر.
- يجب ربط المؤهل t_{fp} بهذه التبعية، بحيث يُشير إلى ان الحالة "يشرب دخاناً" قد توقفت.



مميزات التبعية التصورية:

- إستعمال الحالات الابتدائية يقلل قواعد الإستدلال مما يقلل حجم وزمن تنفيذ المهام المطلوبة.
- العديد من قواعد الإستدلال بشكل آلي ممثلة في تراكيب التبعية التصورية.
- الفتحات في التراكيب الأولية تساعد على التركيز على النقاط التي لم تُؤسَس بعد.

سلبيات التبعية التصورية:

- المعرفة يجب أن تُحلل إلى الحالات الابتدائية وبمستويات منخفضة جداً.
- في بعض الحالات يكون من المستحيل أو صعب جدا إيجاد مجموعة صحيحة من الحالات الابتدائية.
- التمثيل يمكن أن يكون معقداً بالنسبة لبعض الأعمال البسيطة. لنفترض:
زيد يتأهن مع عمر على 100 ريال على ان البرازيل ستربح كأس العالم بكرة القدم.
تتطلب التمثيلات المعقدة الكثير من التحزني

تطبيقات التبعية التصورية:

(Meaning Analysis, Response Generation and Inference on English) **MARGIE**

-- فهم نموذج لغة طبيعي.

(Script Applier Mechanism) **SAM**

-- مخطوطة لفهم القصص.

(Plan Applier Mechanism) **PAM**

(خطة Applier آلية) -- مخطط البرنامج الخاص بفهم القصص.

المخطوطات

المخطوطة هي تركيب البيانات الذي يصف مجموعة الظروف المتوقعة والتي تخولنا المتابعة من مرحلة لإخرى. هي مشابهة لفكرة السلسلة أو سلسلة الحالات التي يمكن أن نتوقع. كما انها يمكن أن تشمل عدد من الشقوق أو الإطار لكن بقواعد أكثر تنظيماً.

المخطوطات مفيدة لأن:

- تقع الأحداث في المخطوطات بمرات أو أنماط معروفة.
- في المخطوطات نجد العلاقات سببية بين الأحداث.
- في المخطوطات نجد شروط الدخول تسمح بوقوع الحدث.
- في المخطوطات نظهر الشروط عندما تقع الأحداث. ومثال على ذلك: - عندما يتقدم طالب ما بدرجته في مخطط الدرجات أو عندما شخص ما يشتري بيت.

مكونات المخطوطة تتضمن:

- شروط الدخول
- شروط الدخول او البدء يجب أن نتحقق قبل وقوع الأحداث في المخطوطة.
- النتائج
- الشروط التي يجب ان تكون صحيحة بعد وقوع الأحداث في المخطوطة.
- الدعائم
- الشقوق التي تمثل الأجسام المشتركة في الأحداث.
- الأدوار
- الأشخاص المشتركين في الأحداث.
- المسار
- الاختلافات على المخطوطة. المسارات المختلفة قد تشترك في مكونات نفس المخطوطة.
- المشاهد
- سلسلة الأحداث التي تحدث. الأحداث ممثلة في شكل التبعية التصوري.

المخطوطات مفيدة في وصف بعض الحالات مثل سرقة مصرف. هذا قد يتضمن:

- الحصول على سلاح.
- احتجاز المصرف.
- الهروب بالمال.
- هنا الدعائم قد تكون
- بندقية، G .
- الغنيمية (المال)، L .
- الحقيقة، B
- الهروب بالسيارة، C .
- الأدوار قد تكون:
- سارق، S.

- صراف، M .
 - مدير المصرف، O .
 - الشرطي، P .
 - شروط الدخول قَدْ تَكُونُ:
 - S فقيرٌ.
 - S مُعْدَمٌ.
 - النتائج قَدْ تَكُونُ:
 - S عِنْدَهُ مَالٌ أَكْثَرُ.
 - O غاضبٌ.
 - M في حالة من الذهول.
 - P يطلق النار.
- هناك 3 مشاهد: الحُصُول على البندقية، سُرُوق المصرف والهروب.
- المخطوطة الكاملة يُمكن أَنْ تُوصَفَ في الرسم 6.

Script: ROBBERY	<i>Track: Successful Snatch</i>
Props: G = Gun, L = Loot B= Bag, C = Get away car.	Roles: R = Robber, M = Cashier, O = Bank Manager, P = Policeman.
Entry Conditions: R is poor. R is destitute.	Results: R has more money. O is angry. M is in a state of shock. P is shot.
Scene 1: Getting a gun R PTRANS R into Gun Shop R MBUILD R choice of G R MTRANS choice. RATRANS buys G (go to scene 2)	
Scene 2 Holding up the bank R PTRANS R into bank R ATTEND eyes M, O and P R MOVE R to M position R GRASP G R MOVE G to point to M R MTRANS "Give me the money or ELSE" to M P MTRANS "Hold it Hands Up" to R R PROPEL shoots G P INGEST bullet from G M ATRANS L to M M ATRANS L puts in bag, B M PTRANS exit O ATRANS raises the alarm (go to scene 3)	
Scene 3: The getaway M PTRANS C	

الرسم. 16 مخطوطة سرقة مصرف

بعض الملاحظات الإضافية حول المخطوطات:

- بحال تطبيق مخطوطة معينة فإنه يجب أن يفعل أولا والتنشيط او التفعيل يعتمد على المعنى.
- في حال ان موضوع المخطوطة مذكور في مخطط سير الحل، فإن المؤشر إلى تلك المخطوطة يُمكن أن يوقف.
- في حال ان الموضوع يحمل صفة مهم فإن المخطوطة يجب أن تُفتح.
- يكمن الخطر في وجود الكثير من المخطوطات (البريمجات) النشيطة في آن واحد او ان يكون له الكثير من النوافذ المفتوحة على الشاشة أو الكثير من النداءات التكرارية في البرنامج.
- نسخ مختلفة قد تُسمح لنتائج مختلفة من المخطوطات (ومثال على ذلك: - سرقة المصرف بحال الفشل).

مميزات المخطوطات:

- امكانية توقع الأحداث.

- مفسر متماسك وحيد قد ينشأ من مجموعة الملاحظات.
- الهليبيات:

- تعتبر اقل شمولية من الإطار.
- قد لا تكون مناسبة لتمثيل كل أنواع المعرفة.

الموسوعة (CYC) EnCYClopedia

ماهي الموسوعة؟

- محاولة طموحة لتشكيل قاعدة معرفة كبيرة جداً تمتلك خصائص الاستنتاج الادراكي.
- الهدف الأول لهذه المحاولة كان استنباط المعرفة من مائة مقالة مختارة بشكل عشوائي من الموسوعة البريطانية (Britannica).
- كل من نوعي المعرفة الضمني والواضح تم تشفيره.

المثال على المعرفة الضمنية والواضحة: لرقترض اننا قرأنا ان زيد علم بموت عمر اذن نحن لبشر يمكن ان نستنتج ان عمر ابدأ لن يعرف بموت زيد.

كيف استنتجنا هذا؟

تطلب الامر فقط معرفة ضمنية خاصة واستنتاجات مثل:

- يهوت الانسان مرة واحدة فقط.
- يبقى الانسان ميتاً اذا مات.
- لا يستطيع الميت العلم بأي شيء.
- الزمن لا يستطيع الرجوع إلى الوراء.

لماذا نثني قواعد المعرفة الكبيرة:

الهشاشة

-- قواعد المعرفة المتخصصة تعتبر هشة. بسبب صعوبة تفسير الحالات الجديدة والضعف في الأداء. قواعد المعرفة المبنية على الادراك يجب ان يكون لديها اسس أقوى.

الشكل والمحتوى

-- تمثيل المعرفة قد لا يكون مناسباً لأنظمة الذكاء الاصطناعي. لكن إستراتيجيات الادراك يمكن ان تؤدي إلى الصعوبات في المحتوى والتي قد تؤثر على الشكل.

المعرفة المشتركة

-- يجب ان يسمح لاتصال أكبر بين الأنظمة بالقواعد والفرضيات المشتركة.

كيف نشفر الموسوعة؟

- بشكل يدوي.
- باستخدام لغة سي واي سي إل الخاصة والتي لها الصفات التالية:
 - o لغة شبيهة بلغة LISP.
 - o تستخدم نظرية الإطار
 - o تستخدم نظرية الميراث المتعدد
 - o تستخدم نظرية الشقوق.
 - o تستخدم الميراث المعمم -- أي ابط ليس فقط المضمون والحالة.

الأنظمة الخبيرة

مقدمة

حتى الآن نحن ناقشنا كيف يمكن ان نتمثل معرفة، لكننا لم نتطرق الى مسألة كيف يمكن ان نستعمل هذا لحل مشاكل عملية حقيقية. هذا القسم سيبحث في كيفية استخدام البعض من التقنيات التي نوقشت حتى الآن في الأنظمة الخبيرة –

والتي يمكن تعريفها على انها : الأنظمة التي تُزوّدنا بنصائح نوعية خبيرة، كذلك تشخّصُ وتوصيات لمشاكل من العالم الحقيقي.

الأنظمة الخبيرة تصمم لحلّ مشاكل حقيقية والتي عادة تُطلَبُ تَخَصُّص و خبرة بشرية (مثل الطبيب أو خبير المعادن). بناء نظاماً خبيراً يتضمّن أولاً إستخراج المعرفة ذات العلاقة من الخبير البشري. هذه المعرفة في أغلب الأحيان لها طابع إرشادي في الطبيعة، وتكون مستندة على البيانات المفيدة ``قواعد الإبهام`` بدلاً من الحقائق المُطلقة. الحصول على المعلومات من الخبير بطريقة ما ليُكوّن ممكن استعملها بعد ذلك من قبل حاسوب تعتبر عموماً مهمة صعبة، وتطلّب خبرته الخاصة. تعتبر مهمة مهندس المعرفة هي إنتزاع المعرفة وبناء قاعدة المعرفة التي تمثل اساس النظم الخبيرة.

أي محاولة أولى في بناء نظام خبير من غير المحتمل أن تكون ناجحة جداً. ذلك وبشكل جزئي لأن الخبير من الصعب بمكان إيجاد المعرفة والقواعد المستخدمة من قبل البشر لحلّ نفس المشكلة المراد مكنتها. وذلك لأن معظم هذه القواعد لا شعوريّاً تقريباً، أو تبدو سهلة جداً للبشر لدرجة انهم لا يفكرون بطريقة اتمامها، ومثال على ذلك حفظنا لوجوه من نعرف. إكتساب المعرفة للأنظمة الخبيرة م جال واسع من البحث، مع نوع عريض ومختلف من التقنيات المطوّرة. على أية حال، عموماً من المهم تطوير نموذجاً أولياً مستند على المعلومات المنتزعة من مُقابلة الخبير، ثمّ تنقيتها وتعديلها بشكل تكراري بالاستناد على آراء كل من الخبير والمستعملين المحتملين للنظام الخبير.

لكي يتمّ مثل هذا التطوير التكراري لنموذج مبدئي، من المهم ان يكون النظام الخبير مكتوب بطريقة بحيث أنه يُمكن اختبارها وتعلّمها بسهولة. النظام يجب أن يُكوّن قادر على توضيح إستنتاجاته (إلى الخبير والمستعمل ومهندس المعرفة) والاجابة على الأسئلة حول عملية الحلّ. تجديد النظام يجب أن لا يتضمّن إعادة كتابة الكثير من اجزاء البرنامج (الكود) - فقط إضافة أو حذف بعض مواضع اجزاء البرنامج.

إنّ مخطّط تمثيل المعرفة المستعمل على نحو واسع للأنظمة الخبيرة هو مجموعة قواعد (وأحياناً بالتمازج مع أنظمة الإطار). نموذجياً، القواعد لن تعطي الإستنتاجات الاكيدة - بل سيُكوّن لدينا فقط بعض درجات اليقين التي ستؤدي بالاستنتاج الى التوقف في حال ان الشروط ادت الى توقف. التقنيات الإحصائية تُستعمل لتقرير هذه الحقائق. النظم المبنية على اساس القواعد، مع أو بدون الحقائق، تعتبر قابلة للتعديل بسهولة عموماً ويجعل الأمر سهلاً لمتابعة وتعقب طريقة التفكير والاستنتاج في هذه النظم. ان اثار هذا التعقب يُمكن أن تُستعمل في اعطاء تفسيرات لآلية عمل النظم.

الأنظمة الخبيرة تُستعمل لحلّ مجموعة واسعة من المشاكل في المجالات العديدة مثل الطبّ والرياضيات والهندسة وعلم طبقات الأرض وعلم الحاسبات والعمل والقانون والدفاع والتعليم.. الخ. ضمن كل مجال، تستعمل الأنظمة الخبيرة لحلّ مشاكل من أنواع مختلفة. بعض المشاكل تتضمّن:

- التشخيص (ومثال على ذلك: -، عيب نظام ما، خطأ الطالب أو تحديد المرض)؛
 - التصميم (أنظمة الحواسيب، الأنظمة الفندقية... الخ)؛
 - التفسير (على سبيل المثال، البيانات الجيولوجية).
- التقنية الملائمة لحل المشكلة تعتمد أكثر على نوع المشكلة من مجال المشكلة. كل الكُتب في هذا الموضوع كُتبت حول طريقة إختيار تمثيل المعرفة وطرق الاستنتاج التي تعطي خصائص المشكلة.

تصميم النظم الخبيرة

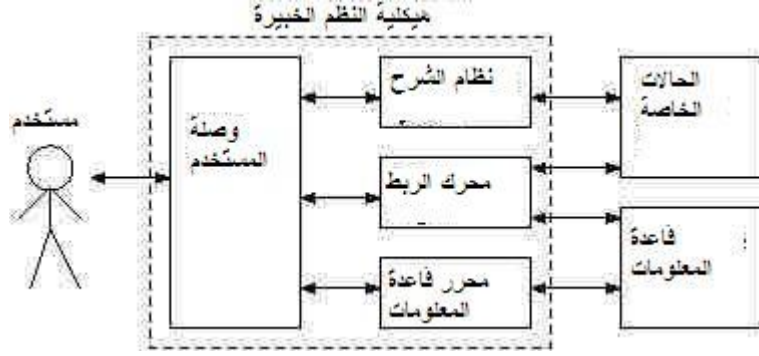
في هذا القسم سنتعرف بتفصيل أكبر على كيفية تصميم وكتابة الأنظمة الخبيرة. أولاً مراجعة وتنقيح المخططات المعمارية الأساسية للنظام الخبير، ثم نحن سنناقش كيفية اختيار أنظمة خبيرة ملائمة لمشاكل محددة وإختيار أنظمة الطّوي.

المخططات المعمارية للأنظمة الخبيرة

الرسم 18 يبين الوحدات الأكثر أهمية التي تتوفر في الأنظمة الخبيرة المبنية على أساس القواعد وهي التالية :

- وصلة التفاعل مع النظام : وهي الوصلة التي يتفاعل من خلالها المستخدم مع النظام والتي يمكن ان تكون :
 - القوائم
 - لغة طبيعية
 - أو أي أسلوب آخر من التفاعل.
- محرّك الإستدلال: والذي يُستعمل للتفاهم مع كل من:
 - المعرفة التخصصية (التي تم الحصول عليها من الخبير).

- البيانات الخاصة بالمشكلة المعيّنة المراد حلها. المعرفة التخصصية نموذجياً ستكوّن على شكل مجموعة من قواعد إذا - آنذاك (IF-THEN).
- بيانات الحالة المعيّنة: وتتضمّن كل من :
 - البيانات التي زوّدت بمساعدة المستخدم.
 - الإستنتاجات الجزئية (السوية مع إجراءات الحقيقة) المستندة على هذه البيانات. في نظام بسيط و محدد مبني على أساس القواعد، بيانات الحالة المعيّنة ستكوّن العناصر في الذاكرة الفعالة.



الرسم 17. هيكلية النظم الخبيرة

- نظام التفسير الفرعي : تقريباً كلّ الأنظمة الخبيرة لها نظام تفسير فرعي، و الذي يسمّح للبرنامج بتوضيح استنتاجاته للمستخدم.
- محرر المعرفة الاساسي : بعض الأنظمة لها أيضاً محرر المعرفة الأساسي الذي يساعد مهندس المعرفة أو الخبير لتحديثه بسهولة ولتوثيق قاعدة المعرفة.
- ميزة مهمة واحدة للأنظمة الخبيرة وهي الطريقتان التي يقوم بها النظام الخبير (عادة) بتقسيم مجال واسع وعام من تقنيات الاستنتاج والتمثيل الى مجالات اصغر لمعرفة محددة . إنّ مجموعة الهدف العام والمجموعة (في الصندوق المنقّط في الرسم) تسمى هيكل النظم الخبيرة. كما نرى في الرسم، الهيكل سريّ ود النظام بالتالي:
 - مُحرك الاستدلال
 - مخطط تمثيل المعرفة
 - وصلة المستخدم
 - نظام التفسير
 - وأحياناً محرر المعرفة الأساسي.
- لنأخذ نوع جديد من المشاكلة المطلوب حلها (مثلاً، تصميم سيارة)، نحن يُمكن أن نجد عادة الهيكل (المخطط المعماري) الذي سيعطينا الخيار الصحيح لدعم حل تلك المشكلة، لذا كلّ ما نحن نحتاجه هو التزوّد بالمعرفة التخصصية. هناك نظم خبيرة عديدة تجارية، كلّ واحد منها ملائم لنوع مختلف قليلاً من المشاكل. استعمال مبدأ الهيكل لكتابة الأنظمة الخبيرة تُخفّض الكلفة عموماً كثيراً ووقت التطوير.

إختيار المشكلة

- تطوير نظاماً خبيراً عموماً تتطلب الكثير من الوقت والمال. لتفادي التكلفة المرتفعة وحالات الفشل، طور بعض الخبراء مجموعة التعليمات (الدليل) للتحديد اي نظم خبيرة مناسبة لحل المسائل والمشاكل المحددة، حسب التالي:
 - الحاجة لحلّ مشكلة ما باستخدام النظم الخبيرة يجب أن يُبرّر العكس التي ستفق على التطوير. لابد أن يكون هناك تقييم واقعي من العكس والمنافع (دراسة جدوى اقتصادية).
 - الخبرة الإنسانية ليست متوفرة في كلّ الحالات المطلوبة. إذا المعرفة التخصصية متوفرة بشكل واسع هذا لايعني بأنها ستؤدي حتماً الى تطوير نظام خبير. على أية حال، في مجالات مثل إستكشاف النفط والطب هناك قد تكون المعرفة المتخصصة نادرة والتي يُمكن أن نحصل عليها بسعر رخيص من نظام خبير، وذلك عندما نحتاج لها، بدون الحاجة لدفع تكاليف كبيرة للخبير نفسه (الشخص).

- المشكلة قد تُحلُّ باستعمال تقنيات التفكيك الرمزية. بحيث أن لا يتطلب مهارة يدوية أو مهارة طبيعية.
 - إن المشكلة مُنظمة بشكل جيد ولا تتطلب الكثير من معرفة الحس العام. معرفة الحس العام صعبة جداً للتمثيل. ويبدو ان التعامل مع حقول بتقنية عالية أسهل ، ويميل إلى تضمين كميات صغيرة نسبياً من المعرفة المُشكلة الجيدة.
 - المشكلة لا يمكن أن تحل بشكل سهل باستعمال طرق الحساب التقليدية. بحيث انه إذا كان هناك من حل حسابي جيد للمشكلة، فإنه تنتفي الحاجة إلى استعمال نظام خبير لحلها.
 - ضرورة توفر الخبراء المتعاونون والواضحو. بحيث ان من اهم عناصر مشروع النظام الخبير الهامج توفر الخبراء الواعيون بالمساعدة، والمستقرون بعملهم ! كما نحتاج لإدارة ومستخدمين محتملين لديهم مواقف إيجابية إلى المشروع بأكمله.
 - إن تكون المشكلة من الحجم والمجال الصحيح. نموذجياً نحن نحتاج إلى المشاكل التي تتطلب خبرة عالية من التخصص، والتي تتطلب من الخبير البشري وقت قصير لحلها (مثلاً ساعة على الاغلب).
- يجب أن يكون من الواضح ان حيز صغير من المشاكل ملائم لتقنية النظم الخبيرة. على أية حال، بوجود مشكلة مناسبة، الأنظمة الخبيرة يمكن أن تعطي منافع هائلة. تم تطوير بعض الأنظمة، على سبيل المثال، للمساعدة على تحليل العينات التي تجمع في عملية استكشاف النفط، وللمساعدة على تشكيل أنظمة الحاسوب. هذه الأنظمة، توفر الكميات الكبيرة من المال، في الاستعمال الصحيح لها.

هندسة المعرفة

بعد أن تبين بأن المسألة او المشكلة مناسبة لأن تحل بواسطة نظام خبير فأنا نحتاج للحصول على المعرفة من الخبير وتمثيلها باستخدام هيكل النظم الخبيرة. هذه وظيفة او مهام مهندس المعرفة، بالتعاون المستمر مع الخبير /المستخدمين.

إن مهندس المعرفة هو خبير تمثيل وبرمجة انظمة الذكاء الاصطناعي. هو يجب أن يكون قادراً على اجراء التالي :

- اختيار هيكل النظم الخبيرة المناسب (أدوات أخرى) للمشروع.
 - الاستحصال على المعرفة من الخبير.
 - تطبيق المعرفة في قاعدة معرفة صحيحة وكفوءة.
- قد لا يكون لدى مهندس المعرفة اي معلومات او معرفة في مجال التطبيق فيما يخص هذه المسألة او المشكلة. لإنتراع المعرفة من الخبير، يجب على مهندس المعرفة اولا أن يلم بعض الشيء على الأقل بمجال المشكلة، لربما بقراءة النصوص التمهيدية أو يتكلم مع الخبير بشكل عام حول الافكار الاساسية. بعد هذا، يقوم بمقابلات أكثر تنظيماً مع الخبير. نموذجياً الخبراء يضربون سلسلة من الامثلة حول المشكلة، وسيوضحون طريقة تفكيرهم وآرائهم في حل هذه المشكلة. مهندس المعرفة سيخرج بقواعد عامة من هذه التفسيرات، ويدققهم مع الخبير.
- كما في أكثر التطبيقات، النظام يعتبر بلافائدة إذا لم يكن المستخدم سعيداً به، لذا فإن التطوير يجب أن يتضمن تعاون كامل مع المستخدمين المحتملين لاحقاً للنظام. كما هو مذكور في المقدمة، دورة التطوير الأساسية يجب أن تتضمن التطوير السريع لنموذج أولي واختبار تكراري وتعديل ذلك النموذج مع كل من الخبراء (لتدقيق صلاحية القواعد) والمستعملون (للتدقيق بأن النظام يمكن أن يزودهم بالمعلومات الضرورية، وبأنهم راضون عن أداء وتفسيرات الأنظمة، وبأنه في الحقيقة يسهل حياتهم بدلاً من ان يعقدها).
- لكي طور النموذج الأولي على مهندس المعرفة أن يتخذ قرارات مؤقتة حول تمثيل المعرفة الملائم وطرق الاستدلال (ومثال على ذلك: -، مبدأ القواعد، أو مبدأ القواعد + الإطار؛ التقييد الأمامي أو الخلفي). لإختيار قرارات التصميم الأساسية، حيث ان النموذج الأول قد يحل فقط جزء صغير من المشكلة العامة. إذا الطرق المستعملة اعطت نتيجة جيدة لذلك الجزء الصغير فإنه من المجدي استثمار الجهد في تمثيل بقية المعرفة في نفس الشكل لحل باقي الاجزاء.

تطوير النظم الخبيرة كان عصرياً جداً في نهايات القرن الماضي، بالتوقعات الغير واقعية حول المنافع المحتملة. الآن تم حل بعض التهكم. هياكل النظم الخبيرة الان تستعمل بشكل واسع جداً، لكن في أغلب الأحيان تستعمل لحل المشاكل البسيطة جداً، ومختارة بسبب وصلة المستخدم بها وبيئات تطويرها وليس بسبب قدراته الإستنتاجية.

القواعد والأنظمة الخبيرة

في هذا القسم سنرى كيف تعمل الأنظمة الخبيرة المبنية على اساس إذا-فإن (IF-THEN) ، و، سنورد مثالا على نظام خبير بسيط جداً باستخدام لغة البرمجة Prolog.

الأنظمة المبنية على أساس القواعد يُمكن أن تُكوّن هدف موجه باستعمال النقيض الخلفي لإختبار بعض الفرضيات ان كانت صحيحة ، أو بيانات موجهة، باستعمال النقيض الأمامي للتوصّل إلى الإستنتاجات الجديدة من البيانات الأصلية. الأنظمة الخبيرة قد تستعمل واحدة أو كلتا الإستراتيجيات، لكن الأكثر شيوعاً استعمال إستراتيجية الهدف الموجه خلفاً. أحد الأسباب لهذا بأنّ النظام الخبير يجب أن يجمع معلومات حول المشكلة من المستخدمين بسؤالهم أسئلة - باستعمال إستراتيجية الهدف الموجه فقط نحن يُمكن أن نَسأل الأسئلة التي لها علاقة بالحل المقترح. على أية حال، في النظم الخبيرة المبنية على اساس القواعد - الهدف الموجه البسيط هناك في أغلب الأحيان مجموعة الحلول المحتملة للمشكلة - كمثال يمكن اعتبار مجموعة الأمراض التي من المحتمل ان تكون عند مريض ما. النظام الخبير سيُعتبر كل حل مُقترح (ومثال على ذلك: -، عنده نزلة برد (فريد) (has_cold(fred))) ويُحاول إثبات إذا ما كان هو الحالة أو لا. أحياناً لا يكون النظام قادراً على إثبات أو تفنيد الشيء من البيانات الأولية المجهزة من قبل المستخدم، لذا هو سيُسأل المستخدم بعض الأسئلة (ومثال على ذلك: -، "هل أنت تشعر بصدا؟"). استعمال البيانات الأولية بالإضافة الى الاجابة على الأسئلة يجب أن يمكن النظام من إستنتاج اي من الحلول المحتملة لهذه المشكلة يعتبر الصحيح.

مثال بسيط

سيُوضّح هذا بشكل أفضل من خلال مثال بسيط. لِنَقْرَضْ بأنّ لدينا القواعد التالية:

1. IF engine_getting_petrol
AND engine_turns_over
THEN problem_with_spark_plugs
2. IF NOT engine_turns_over
AND NOT lights_come_on
THEN problem_with_battery
3. IF NOT engine_turns_over
AND lights_come_on
THEN problem_with_starter
4. IF petrol_in_fuel_tank
THEN engine_getting_petrol

مشكلتنا هنا أنّ نعرف ما هو العطل في السيارة التي أعطت بعض الأعراض الجديرة بالملاحظة. هناك ثلاث مشاكل محتملة بالسيارة:

- مشكلة ضعف شرارة التشغيل.
- مشكلة بالبطارية.
- مشكلة ببدائ التشغيل.

سنفترض بأننا لم نكن مجهزون بالحقائق الأولية حول الأعراض الجديرة بالملاحظة.

في نظام الهدف المُوجّه الأسهل والذي نحن نحاول إثبات كل مشكلة مُفترضة (بالسيارة) تبعاً. أولاً النظام يُحاول إثبات "مشكلة ضعف شرارة التشغيل". القاعدة 1 افتراضياً مفيدة هنا، لذا النظام يَصِغ الأهداف الجديدة للإثبات "المحرك يحصل على البنزين" و "المحرك يبدأ بالدوران ولا يعمل". بمحاولة إثبات الأولى من هذه، القاعدة 4 يُمكن أن نستعمل، مع الهدف الجديد من الإثبات "البنزين في خزان الوقود" ليس هناك قاعدة التي تستنتج هذا (والنظام لا يعرف الجواب)، لذا النظام سيُسأل المستخدم: هل صحيح أن هناك بنزين في خزان الوقود؟

لنقول ان الجواب نعم. هذا الجواب سَيُسْجَلُ، لكي لا يُسْأَلَ المستخدم نفس السؤال ثانيةً. على أية حال، أثبتَ النظام الآن بأن المحركَ يَحْصُلُ على البنزين، لذا يُريدُ الآن أن يكتشفَ إذا المحركَ يبدأ بالدوران. بما ان النظام لا يَعْرِفُ لحد الآن ان هذه هي الحالة، وبما انه ليس هناك قواعد التي يمكن ان تُستنتجَ هذا، سيقوم بسؤال المستخدم: هل هو صحيح ان المحرك بدأ بالدوران؟

لنفترض ان الجواب لا. ليس هناك قواعد أخرى التي يمكن أن تُستعملَ لإثبات `` مشكلة _ ضعف شرارة التشغيل " لذا النظام سَيستنتجُ بأن هذا ليس الحلَّ لهذه المشكلة، وسيعتبرُ الفرضيةَ القادمة: مشكلة _ مع البطارية. صحيح ان المحرك لم يبدأ بالدوران، حسب جواب المستخدم ، لذا هو يَجِبُ أن يُثبِتَ بأن ال مصابيح لا تضيئ. هو سَيَسْأَلُ المستخدم:

هل صحيح ان المصابيح تضيئ؟

لنفترض ان الجواب لا. أثبتنا الآن بأن المشكلة بالبطارية. بعض الأنظمة قد تتوقفُ هناك، لكن عادة هناك قد يكون أكثر من حل واحد، (ومثال على ذلك: -، أكثر من عيب واحد بالسيارة)، أو سيكون الحل الصحيح مجهولاً من بين الحلول المختلفة. لذا كل الفرضيات مُعتبرة. هو سَيحاولُ إثبات `` مشكلة _ مع مبدئ التشغيل "، لكن أعطى البيانات الحالية (المصابيح تضيئ) البرهان سَيَفْشَلُ، لذا النظام سَيستنتجُ بأن المشكلة بالبطارية. التفاعل الكامل مع نظامنا البسيط جداً قد يكون:

النظام: هل صحيحاً بأن هناك بنزين في خزان الوقود؟

المستخدم: نعم.

النظام: هل صحيحاً بأن المحرك يبدأ بالدوران؟

المستخدم: لا.

النظام: هل صحيحاً ان المصابيح تضيئ؟

المستخدم: لا.

النظام: أستنتج بأن هناك مشكلة بالبطارية.

نلاحظ انه عموماً، حل المشاكل باستعمال التوقييد الخلفي تتضمن التفتيش خلال كل الطرق المحتملة لإثبات الفرضية، وبشكل منظم يُدَقَّقُ كل واحد منهم. الطريق المشترك من عمل هذا البحث يشبه تماماً مثل في Prolog – بحث العمق أولاً بالتراجع.

وسائل التفسير

أكثر الأنظمة الخبيرة لديها وسائل تفسير التي تسمَحُ للمستخدم بطرح بعض الأسئلة عن سبب السؤال التي تطرحه هي بدورها، وعن كيفية توصلها لبعض النتائج. هذه الأسئلة مُجابهة بالرجوع إلى اهداف النظام ، القواعد تبدأ بلأن تكون مستعملة، وتخرج بيانات المشكلة. لتوضيح أنواع الوسائل المجهزة عموماً، نحن سنتوسع على مثالنا البسيط المعطى اعلاه، بدء مع الحوار في المثال البسيط والذي يتضمن أسئلة وتفسيرات مثل "لماذا" و "كيف".

النظام: هل هناك بنزين في خزان الوقود؟

المستخدم: نعم.

النظام: هل صحيح ان المحرك بدأ بالدوران؟

المستخدم: لا.

النظام: هل ان المصابيح تضيئ؟

المستخدم: لماذا؟

النظام: أحاول إثبات: ان المشكلة بالبطارية.

هو أسس بأنها ليست الحالة التي تقول ان المحرك بدأ بالدوران.

لذا إذا هي ليست الحالة التي تخبر ان المصابيح تضيئ فإنه هناك مشكلة بالبطارية.

هل هو صحيح ان المصابيح تضيئ؟

المستخدم: لا.

النظام: أستنتج بأن هناك مشكلة بالبطارية.

المستخدم: كيف؟

النظام: هذا يتلّى من قاعدة 2:

IF NOT engine_turns_over AND ...

NOT engine_turns_over was given by the user.

NOT lights_come_on was given by the user.

ادراج وسيلة التفسير تتضمن، على الأقل، شجيرة ماذا تستعمل القواعد في تجهيز الاستنتاجات، ويستعمل هذه السجلات لإعداد التفسيرات مثل الواردة في المثال المعطى أعلاه. إعطاء التفسيرات البسيطة، لثلك في نفس المثال، ليس صعباً جداً (الأجوبة دائماً لها شكل مجموعة بسيط)، ومفيد جداً أحياناً. على أية حال، وسائل التفسير في الأنظمة الخبيرة في أغلب الأحيان ليست كثيرة الإستعمال، و إذا إستعملت فهي لاتبدو مقبولة من قبل المستخدمين. هناك الكثير من الأسباب لهذا، مما يحفز البحث الآن في المنطقة. أحد هذه الأسباب لان التفسيرات فقط ترجع المعرفة السطحية المشفرة في القواعد، بدلاً من المعرفة العميقة حول المجال الذي حفز القواعد أصلاً (والتي بالاصل ليست مُمثلة). لذا، النظام سيَقول بأن إستنتاج إكس بسبب القاعدة 23، لكن لا يوضح كل شيء عن القاعدة 23. (في المثال أعلاه، يحتاج المستخدم لفهم ان كل من المصابيح ومبدئ التشغيل يستعملان البطارية، التي تعتبر السبب الجوهري للقاعدة الثانية في مثالنا). السبب المعلن الآخر للفشل المتكرر لوسائل التفسير هو الحقيقة بأن، إذا أخفق المستخدم في فهم أو قبول التفسير، فإن النظام لا يستطيع إعادة التوضيح بشكل آخر (لما يستطيع الانسان ان يفعل). توليد التفسيرات يعتبر منطقة بحث كبيرة جداً، مُهتمة بالإتصال الفعال: كيف نمثل الأشياء بحيث ان الفاس يكونون راضون عن التفسير، وما نتائج ذلك هل تمثل المعرفة التحتية.

الأنظمة الأكثر تعقيداً

القواعد الأربع في المثال أعلاه تُصور الأفكار الأساسية للأنظمة الخبيرة المبنية على أساس القواعد ذات الهدف الموجه. على أية حال، عموماً بالطبع هناك سيكون مئات أو آلاف القواعد في النظام، وكل قاعدة ستكون أكثر تعقيداً إلى حد كبير. القواعد ستتضمن الأنماط بالتأكيد تقريباً بالمتغيرات (ومثال على ذلك: (age(Patient, X) - عمر (مريض، إكس))، بدلاً من مقترحات غير منظمة بسيطة (ومثال على ذلك: (age_fred_23)) والتي عادة لديها عوامل حقيقة ارتبطت بهم (سيتم وصفها لاحقاً).

على أية حال، بالرغم من أن الأنظمة الحقيقية ستكون أكثر تعقيداً من تلك الموصوفة أعلاه، حيث تستعمل الكثير من نفس التقنيات والاستنتاج الأساسي وإجراءات التفسير. نظام مبكر واحد إستعمل هذه النظرية الأساسية ويدعى MYCIN، وهو نظام لتشخيص الإضطرابات الدموية. نحن سنصنف هذا النظام بشكل مفصل أكثر بعض الشيء لاحقاً.

هيكلية نظام خبير في Prolog

إن البرنامج التالي هو هيكلية نظام خبير بسيطة جداً في Prolog. نؤرده لنرى كيف يمكن ان تبدو الهيكلية الأساسية، ولتقديم بضعة ميزات جديدة للغة البرمجة Prolog.

نلاحظ انه لكي نعرف النحو الخاص للقواعد نستعمل بعض إعلانات المشغل (: op .). هذا يسمح لنا أن يكون عندنا تعابير مثل ``if small(X) and nice(X) then good(X) إذا إكس صغير ولطيف يعني اكس جيد ". على أية حال، هذه حقاً فقط حقائق Prolog، التي يمكن أن تكون نمطاً مقارناً على نحو طبيعي. لتكوين فكرة عن هذا، حاول كتابة وتشغيل هذا البرنامج واختبر الإستفسار التالي:

```
?- (if a(A) and B then C) = (if a(1) and a(2) then a(3)).
```

على أية حال، هنا الهيكلية الصغيرة.

```
%% Micro Expert System.
```

```
%% This is a very simple backward chaining rule-based
%% expert system. Given some possible hypotheses it backward chains on
%% each one trying to find out if it is true or not. If it cant
%% prove a fact using the rules it will just ask the user if it is
%% true or not. Once it has proved one of the hypotheses it will
present
```

```

%% its conclusions to the user.

% -----
% OPERATOR DECLARATIONS

:- op(975, fx, if).
:- op(950, xfy, then).
:- op(925, xfy, and).

% -----
% EXPERT SYSTEM RULES (EXPERT KNOWLEDGE BASE)

rule(if eats(X, Y) and living(Y) then carnivore(X)).
rule(if carnivore(X) and big(X) then dangerous(X)).
rule(if has_feathers(X) then bird(X)).
rule(if bird(X) and small(X) then good_pet(X)).
rule(if cuddly(X) then good_pet(X)).

% -----
% TEXT TEMPLATES FOR ENGLISH

% qtext/2: Provides a simple template based translation into English
% questions.

qtext(cuddly(X), ['Is ', X, ' cuddly?']).
qtext(has_feathers(X), ['Does ', X, ' have feathers?']).
qtext(small(X), ['Is ', X, ' small']).

% atext/2: Template based translation into recommendations.

atext(good_pet(X), ['I suggest that ', X, ' would make a good
pet.']).

% -----
% EXAMPLE TEST CASE

find_good_pet :-
    check_hypothesis([good_pet(lenny), good_pet(eddie),
good_pet(tweety)]).

% -----
% MAIN EXPERT SYSTEM SHELL CODE

% check_hypothesis(+Hypotheses)
% Succeeds when one of the hypotheses is proved true, or it
% has tried them all.
% Picks a hypothesis, and uses b_chain to find out if it is true.
% If it is true then b_chain succeeds and check_hypothesis writes out
the
% appropriate recommendation. If false it backtracks to 'on' to find
% another hypothesis to try.
% Once it has tried all the hypotheses it will backtrack to second
% check_hypothesis clause and write an appropriate message.
% (Uses MacProlog built in predicates message/1 and on/2)

check_hypothesis(Hypotheses) :-

```



```

on(Hypoth, Hypotheses),      % get a member of hypotheses
bchain(Hypoth), !,           % b_chain to check if true.
atext(Hypoth, Text),          % get hold of appropriate text.
message(Text).                % write out the recommendation

check_hypothesis(_) :- message(['None of the possible hypotheses seem
to be true']).

% bchain(+Goal)
% Succeeds if Goal is true, given rules + facts supplied by user as
% backward chaining proceeds.

bchain(G1 and G2):- !,        % G1 and G1 are true if
    bchain(G1),                % G1 can be proved by backward
chaining
    bchain(G2).                % and G2 can be too.

b_chain(Goal) :-               % G's true if its a fact!
    userfact(Goal).

bchain(Goal):-                 % Goal is true if
    rule(if Preconditions then Goal), % there's a rule concluding it
    bchain(Preconditions).        % and its Preconditions can be
                                   % proved by backward chaining

bchain(Goal):-                 % Goal is true if
    user_says_its_true(Goal).     % user says its true.

% user_says_its_true(+Goal)
% True if there is some text to use to ask the user about it,
% and when you ask the user they say yes.
% Uses MacProlog built in predicate yesno/1

user_says_its_true(Goal) :-
    qtext(Goal, Text),
    yesno(Text),
    assert(userfact(G)).        % Add the fact to Prolog's
database.

```

ان مسند المستوى الأعلى الرئيسي في Prolog هو مسند فرضية _التدقيق. هذه المسند يستخدم طريقة _الراجع للمرور بكل أعضاء قائمة الفرضيات ل معرفة اي من هذه العناصر يُمكن أن يُثبتوا الصح بالتقييد الخلفي. عند ايّاد عنصر يتبنى الصح، يكتب النظام رسالة إلى المستخدم. لعل هذا هو يبحث أولاً عن بعض النص المخصص لإستنتاج معين (ومثال على ذلك: (tweety) pet _ good)، -، تويتي حيوان اليف-جيد (يستعمل المسند /2 atext، ثم يستعمل مسند الرسائل الموجود ضمناً في الـ Prolog رسالة /1 لكتابة رسالة مناسبة. الرسالة تأخذ حجة وحيدة والتي يجب أن تكون قائمة، وتظهر نافذة ويكتب النود في تلك القائمة.

مسند التقييد الرئيسي b_chainpredicate عند أربعة حالات، يتعامل مع الحالات التي بها:

- ارتباط الأهداف للإثبات.
- الهدف للإثبات فقط انه حقيقة.
- القواعد التي تجاري خاتمتها الهدف (لذا شروطه المسبقة تضع كأهداف جديدة).
- الحقيقة حيث يقول المستخدم بأن الشيء صحيح.

المستخدم يعطي الجواب باختيار نعم او لا " user_says_its_true ' على المسند المبني ضمناً في Prolog yesno /1 الذي يظهر نافذة حوار ببعض النصّ وال مفتاحين (نعم ولا) وتنتج في حال ان المستخدم يقرّ على 'نعم'. إذا يقول المستخدم بأن شيء ما حقيقي فإن تلك الحقيقة مُصرّحة إلى قاعدة بيانات Prolog (التي ستحتوي بعد ذلك على

أشياء مثل ``userfact (لها ريش (tweety)). النصريّجّ او حقائق المستخدم هو مسند خاص في Prolog وذلك للأشياء المضافة بفاعلية إلى قاعدة بيانات prolog. قد نلاحظ الكثير من علامات التعجب ``! " المنتشرة في نص البرنامج اعلاه. والتي تشتمل للسيطرة على التراجع. ملاحظة: ``messge و ``yesno " ليست مسندات قياسية في Prolog. إن القياسي هو write/1 والذي فقط يكتب أي تعبير Prolog، read/1 الذي فقط يقرأ أي تعبير Prolog ويبطه بمتغير. لذا، أعطى الإستفسار read(A)، المستعمل يُمكن أن يطبع، مثلاً، ``fred. " و عندها ترتبط A ب fred. يُرود Prolog بالكثير من: النوافذ الإضافية / القائمة / مفاتيح مسندات الادخال والاخراج. نورد هنا مثال على حوار من النظام أعلاه (وذلك بين : 'S' و 'U': إشارة إلى النظام والمستخدم):

```
?- find_good_pet.
```

```
S: Does lenny have feathers?
U: No.
S: Is lenny cuddly?
U: no
S: Does eddie have feathers?
U: yes.
S: Is eddie small?
U: no
S: Is eddie cuddly?
U: no
S: Does tweety have feathers?
U: yes
S: Is tweety small?
U: yes
S: I suggest that tweety would make a good pet.
```

واضح من الوارد اعلاه انه ليس كافي ليكون هيكل نظام خبير. كبداية هو يسمُح فقط بالاجوبة نعم / لا ؛ لا يسمُح للقواعد التي تتضمن الحقائق؛ و لا يحتوي على وسيلة تفسير. إذا أردنا مكوّن تفسير للنظام فنحن نحتاج لتسجيل مسار القواعد المستخدمة للوصول الى الاستنتاج. كحدّ أدنى نحن يُمكن أن نسجل فقط القواعد التي إستعملت، ثم إذا سأل المستخدم ``كيف؟" فان النظام يُمكن أن يكتب شيء مثل:

إستنتجتُ بأنّ tweety حيوان أليف جيد، وذلك باستعمال القواعد التالية:

```
rule: has_feathers(X) ==> bird(X)
rule: bird(X) & small(X) ==> good_pet(X)
```

حتى الآن أعطينا فكرة عن كيفية كتابة نظام (هيكل) بسيط مبني على اساس القواعد – التقيد الخلفي. عموماً هيكل النظم الخبيرة ستسمح بالتقييد الأمامي أيضاً، وستسمح بتمثيل البعض من المعرفة بلستعمال نظام الإطار. على سبيل المثال، نحن قد نريد أن يكون عندنا مجموعة الإطار التي تقول (بين الأشياء الأخرى) ان روبي أرنب أبيض والأرناب محبوب دائماً. نحن يُمكن أن ندقق بلستعمال التقييد الخلفي لإكتشاف إذا ما روبي يكون حيوان أليف و جيد، لكن التدقيق باستعمال نظام الإطار لإكتشاف إذا ما روبي محبوب (استعمال ميراث من إطار الأرنب).

لمحة عن MYCIN

Mycon وكما هو ورد أعلاه، كان أحد الأنظمة الخبيرة ال مبكرة، وتصميمه أثر على تصميم الأنظمة الخبيرة التجارية بشكل كبير. Mycon كان نظاماً خبيراً طور في ستانفورد في السبعينات. وظيفته كانت تشخيص وإعطاء توصيات حول معالجة بعض إصابات الدم. ليعمل التشخيص، بشكل صحيح، دعم النظام بثقافات متزايدة عن إصابة الكائن الحي. لسوء الحظ هذا يأخذ حوالي 48 ساعة، وإذا إنتظر الأطباء حتى ينجز التشخيص فمن المحتمل ان يموت المريض خلال هذه الفترة! لذا، فإن الأطباء يجب أن يجنبوا بالتخمينات السريعة حول المشاكل المحتملة من البيانات المتوفرة،

وتستعمل هذه التخمينات لتزويد المعالجة بالتوصيات حول اي ادوية يجب ان تُعطي وكيف يمكن التعامل مع أي مشكلة محتملة.

Mycin طور جزئياً لإك شواف كيف ان الخبراء البشريين يعطون تخمينات قاسية (لكن مهمة) بالاستناد على المعلومات الجزئية. على أية حال، المشكلة فعلاً مهمة في الشروط العملية - هناك الكثير من الأطباء الغير مُتخصصين الذين أحياناً يجب أن يعطوا مثل هذا التشخيص القاسي، وإذا كان هناك أداة خبيرة متوفرة لمساعدتهم فإن هذه قد يُمح بمعالجة أكثر فعالية. في الواقع، Mycin لم يستعمل أبداً في الحقيقة عملياً. وهذا لم يكن بسبب أي ضعف في أدائه، بل كانت بسبب أخلاقي ومسائل قانونية تعلقَت باستعمال الحاسبات في الطب - بحيث انه إذا أعطي التشخيص الخاطئ، من تقاضي؟

على أية حال Mycin مثل معرفته لمجموعة قواعد إذا-فإن (IF-THEN) مع عوامل اكيدة. نورد هنا إحدى قواعد Mycin:

IF the infection is primary-bacteremia
AND the site of the culture is one of the sterile sites
AND the suspected portal of entry is the gastrointestinal tract
THEN there is suggestive evidence (0.7) that infection is bacteroid.

0.7 تعتبر تقريباً الحقيقة التي ستُكون النتيجة التي أعطت الدليل. إذا الدليل مجهول، فإن اجزاء حقائق الدليل ستندمج مع حقيقة القاعدة لإعطاء حقيقة النتيجة.

Mycin كُتب باستخدام لغة LISP، وقواعده تُمثل رسمياً تعابير LISP. الجزء الفعلي للقواعد يُمكن فقط أن يُكون النتيجة والتشخيص للمشكلة المراد حلها، أو يُمكن أن تكون تعبير LISP اخر. هذا سمح بمرونة كبيرة، لكن أزال البعض من معايير ووضوح الأنظمة المبنية على أساس القواعد.

على أية حال، Mycin (أولياً) نظام مُوجّه الهدف، يستعمل إستراتيجية التفكير والاستنتاج المقيد خلفياً حسب ما وصفاً سابقاً. على أية حال، إستعمل Mycin العديد من الموجهات للسيطرة على البحث لحل (أو برهان بعض الفرضيات). هذا كان ضروري لجعل التفكير والاستنتاج كفوء ولمنع المستعمل من أن يُسأل الكثير من الأسئلة الغير ضرورية.

Mycin- مع انه اعتبر نظام البحث الخبير و الرائد، كان لديه عدد من المشاكل التي عولجت لاحقاً، بمخططات معمارية أكثر تطوراً. أحد هذه المشاكل كان ان القواعد خلطت معرفة مجال ما في أغلب الأحيان، حل مشكلة المعرفة و فحص شروط " (لتجنب طرح سؤال/ أسئلة سخيفة أو صعبة على المستخدم - ومثال على ذلك: -، التدقيق فيما اذا المريض ليس طفلاً قبل السؤال عن إدمان الخمر). النسخة اللاحقة والتي سميت بـ Neomycin عدلت للتعامل مع هذا بإملاك علم تصنيف أنواع مرض واضح (مثلت المعلومات كنظام إطر) لتمثيل الحقائق حول الأنواع المختلفة من الأمراض. المشكلة الأساسية في إستراتيجية الحل كانت أن النظام ينزل اسفلاً بجميع خيارات شجرة المرض، من الصنف العام من الأمراض إلى الانواع المخصصة جداً، يجمع المعلومات للتفريق بين فئتي المرض الفرعيتين.

كان هناك العديد من التطويرات الأخرى من مشروع Mycin. على سبيل المثال، Emycin كان حقاً الهيكل الخبير المطور من Mycin. نظام خبير جديد اسمه PUFF طور لإستعمال Emycin في مجال اضطرابات القلب. و Neomycin طور لتدريب الأطباء، الذي يأخذهم خلال حالات المثال المختلفة، يُدقق إستنتاجاتهم ويوضح أين فشلوا. نحن يجب أن نوضح نقطة هنا وهي ان ليست كل الأنظمة الخبيرة شبيهة Mycin. يستعمل الكثير من الانظمة الطرق المختلفة لحل نفس المشكلة وتمثيل المعرفة.

الإستنتاج مع الحيرة: التفكير الغير رتيب

ما هو الاستنتاج؟

- عندما نطلب من أي نظام معرفة عمل شيء ما، نحن لم نُخبر بشكل واضح كيف يجب ان يقوم بذلك، هو يجب أن يفكر ويستنتج.
 - النظام يجب أن يفهم ما يحتاجه من المعرفة من نفس مخزون المعرفة لديه.
- رأينا مثال بسيط من التفكير والاستنتاج أو رسم الإستدلالات. على سبيل المثال إذا نحن نعرف أن: ابو الحناء طير. ونعرف ان كل الطيور لها أجنحة. ثم نسأل: هل ابو الحناء له أجنحة؟

ببعض التفكير والاستنتاج (ولو أنه بسيط جداً) يجب أن تكون الإجابة على السؤال نعم.

كيف نفكر ونستنتج؟

إلى حد معين هذا سيعتمد على مبدأ تمثيل المعرفة المختار. مخطط تمثيل المعرفة الجيد يجب أن يسمح بتفكير معقول وطبيعي وسهل. أدرج أسفلاً عدة طرق واسعة جداً عن كيفية تفكيرنا واستنتاجنا. نحن سندرس حالات معينة للبعض من هذه الطرق في الفصول القادمة.

التفكير الرسمي

-- قواعد أساسية من الاستدلال بتمثيل المعرفة المنطقي.

التفكير الإجرائي

-- يستعمل الإجراءات التي تُحدد كيفية حل المشاكل/المشاكل الفرعية.

التفكير بالتناظر الوظيفي

-- يعتبر المبدأ الأكثر صعوبة لأنظمة الذكاء الاصطناعي، بينما نجد أنه سهل جداً للإنسان ومستخدم بكثرة. ومثال على ذلك: - إذا نحن سألنا هل يمكن لأي الحناء أن يطير؟. النظام قد يستنتج بأن أبو الحناء هو عصفور وهو يعرف أن العصفور يمكن أن يطير لذا هو يقرر بأن أبو الحناء يطير.

التعميم والتجريد

-- هذا المبدأ أيضاً فعال بالنسبة للبشر. هذا المبدأ يعتبر أساس طرق التعلم والفهم.

تفكير المستويات المركبة

-- يستعمل المعرفة التي يعرفه ويقوم بطلب نوع المعلومات حسب أهميتها.

التفكير المتردد

كلنا يعرف أن العالم مكان متقلب. أي نظام ذكاء اصطناعي يطلب منه البحث والتفكير والاستنتاج في مثل هذا العالم يجب أن يكون قادر على التعامل مع خاصية الغموض في هذا العالم. بشكل خاص هو يجب أن يكون قادر على التعامل مع:

- النقص -- تعويض النقص في المعرفة.
 - التضارب -- حل حالات الغموض والتناقضات.
 - التغير -- هو يجب أن يكون قادر على تجديد قاعدة معرفته بمرور الوقت.
- بشكل واضح لكي يتعامل مع هذا، نجد أن عدد القرارات التي ستصل إلى حالة "صحيح" (أو خاطئ) هي الأكثر ونحن يجب أن نقدم الطرق التي يمكن أن نتعامل مع هذه الحيرة. هناك ثلاثة طرق أساسية التي يمكن أن تعمل هذا:

- طرق رمزية.
 - طرق إحصائية.
 - طرق المنطق الغير واضح.
- نحن سننظر إلى الطرق الرمزية في هذه الفصل ونشاهد الآخرين في الفصول القادمة.

التفكير الغير رتيب

المنطق المسند والاستدلالات التي قدمناها في المثال سابقاً تدل على التفكير الغير الرتيب. في هذا النوع من التفكير إذا نحن توسعنا في مجموعة الديديات فإننا لا نستطيع التراجع عن أي مزاعم أو بديهيات ناتجة.

بينما نجد أن الإنسان بطريقة تفكيره واستنتاجه لا يلتزم بتراكيب هذا المبدأ عندما يفكر:

- نحن نحتاج لإستباق النتائج لكي نخطط، والأهم لننجو من المخاطر في حال أخذ قرارات سريعة.

o نحن لا نستطيع توقع كل النتائج المحتملة من خطتنا.

o نحن يجب أن نضع الفرضيات حول الأشياء التي لا نعرفها بشكل مُحدد.

الإستنتاج بالتزكية

هذا النوع من الاستنتاج شائع جداً في التفكير الغير رتيب. هنا نريد أن نستنتج القرارات بالاعتماد على قربها للحقيقة.

رَأْيُنَا أَمْثَلَةٌ عَنْ هَذَا النُّوعِ مِنَ الِاسْتِنْتَاكِ وَعَنْ الطَّرِيقِ الْمُحْتَمَلَةِ لِتُمَثِيلِ هَذِهِ الْمَعْرِفَةِ. سَنُناقِشُ هُنَا نَظَرِيَّتَيْنِ لِعَمَلِ ذَلِكَ:

- المنطق الغير الرتيب.
 - منطق التركيبية.
- لَا سُبُوحُ نَفْسِكَ بِالنَّسْبَةِ لِلْمَعْنَايِ "غَيْرِ مُرْتَبِطٍ بِالْمَقْدَمَاتِ" وَ "التركيبية" أَنْ يُطَبَّقَ فِي مَبْدَأِ التَّفَكُّيِّ وَالْمَنْطِقِ الدَّقِيقِ. التَّفَكُّيِّ الرَّتِيبِ يَعْتَبَرُ أَوْصَافُ عَامَّةٌ لِصَنْفٍ أَوْ طَرِيقَةٍ تَفَكُّيِّ. الْمَنْطِقُ الرَّتِيبِ يَعْتَبَرُ نَظَرِيَّةَ نَوْعِيٍّ. نَفْسُ الشَّيْءِ مُمَكِّنٌ قَوْلُهُ عَنْ اسْتِنْتَاكِ التَّرْكِيْبَةِ وَمَنْطِقِ التَّرْكِيْبَةِ.

المنطق الغير رتيب

مَبْدَأِيًّا يَعْتَبَرُ الْمَنْطِقُ الْغَيْرِ رَتِيبِ إِمْتِدَادًا لِمَنْطِقِ مَسْنَدِ الطَّلَبِ الْأَوَّلِ وَذَلِكَ لِتَضْمِينِ مَشْغَلِ نَمَطِي، M . بِغَرَضِ السُّمُوحِ لِلِاسْتِنْتَاكِ عَلَى سَبِيلِ الْمَثَالِ:

$\forall x: \text{plays_instrument}(x) \wedge \text{improvises}(x) \rightarrow \text{jazz_musician}(x)$

الْحَالَةُ الَّتِي تَخْبِرُ أَنَّ كُلَّ x ، حَيْثُ، x يَعْزِفُ عَلَى آلَةٍ مُوسِيقِيَّةٍ، وَإِذَا عَرَفْنَا الْحَقِيقَةَ أَنَّ x يُمَكِّنُ أَنْ يَتَجَلَّ وَانْهِيَ مُتَّسِقَةٌ مَعَ كُلِّ الْمَعَارِفِ الْأُخْرَى، فَأَنَّهُ يُمَكِّنُ أَنْ نَسْتَنْتَجَ أَنَّ x مُوسِيقِيٌّ جَازٍ مَثَلًا. كَيْفَ نَعْرِفُ الْإِتْسَاقَ؟

الْحَلَّ الشَّاعِ الْوَاحِدَ (وَالْمُتَّسِقَ مَعَ رَمُوزِ Prolog) هُوَ أَنَّهُ لِبَيَانِ أَنَّ الْحَقِيقَةَ P صَحِيحَةٌ، يَحَاوِلُ إِثْبَاتِ $\neg P$ فِي P . إِذَا زُتَّوَصَلْنَا بِهَذَا الْإِثْبَاتِ إِلَى "خَطَأٍ" يَعْنِي أَنَّ P مُطَابِقٌ (حَيْثُ عَكْسُهُ خَاطِئٌ). لِنَعْتَبِرُ الْمَجْمُوعَةَ الْمَشْهُورَةَ الْمَتَّعِلَّةَ بِالرَّئِيسِ نِيكْسُونِ.

$\forall x: \text{Republican}(x) \wedge \neg \text{Pacifist}(x) \rightarrow \neg \text{Pacifist}(x)$

$\forall x: \text{Quaker}(x) \wedge \text{Pacifist}(x) \rightarrow \text{Pacifist}(x)$

هَذَا يُصَرِّحُ أَنَّ الْكُويْكِرِيِّينَ يَمِيلُونَ إِلَى أَنْ يَكُونُوا مُسَالِّمِينَ بَيْنَمَا الْجُمْهُورِيِّينَ لَا يَفْعَلُونَ. لَكِنْ نِيكْسُونُ كَانَ كُويْكِرِيٍّ وَجُمْهُورِيٍّ لِذَا نَحْنُ يُمَكِّنُ أَنْ نُصَرِّحَ:

$\text{Quaker}(\text{Nixon})$

$\text{Republican}(\text{Nixon})$

هَذَا يُؤَدِّي إِلَى تَنَاقُضٍ فِي مَعْرِفَتِنَا الْكُلِّيَّةِ.

منطق التركيبية

منطق التركيبية يُقَدِّمُ قَاعِدَةَ اسْتِدْلَالٍ جَدِيدَةٍ:

$$\frac{A \vdash B}{C}$$

الَّذِي يَذْكُرُ أَنَّهُ إِذَا A قَابِلٌ لِلِاسْتِنْتَاكِ وَهُوَ مُطَابِقٌ لِإِفْتِرَاضِ B فَإِنَّهُ مُمَكِّنُ اسْتِنْتَاكِ C . هَذَا مُشَابِهٌ لِلْمَنْطِقِ الرَّتِيبِ لَكِنْ هُنَاكَ بَعْضُ الْإِمْتِيَازَاتِ:

- قَوَاعِدُ اسْتِدْلَالٍ جَدِيدَةٍ مُسْتَعْمَلَةٌ لِحَسَابِ مَجْمُوعَةِ الْإِمْتِدَادَاتِ الْمَعْقُولَةِ. لِذَا فِي مَثَالِ نِيكْسُونِ الْوَارِدِ أَعْلَاهُ، مَنْطِقُ التَّرْكِيْبَةِ يُمَكِّنُ أَنْ يَدْعَمَ كُلُّنَا الْمَزَاعِمَ، حَيْثُ أَنَّهُ لَا يَقُولُ أَيُّ شَيْءٍ حَوْلَ الْإِخْتِلَافِ بَيْنَهُمْ -- هُوَ سَيَعْتَمِدُ عَلَى الْإِسْتِدْلَالِ الَّذِي يَدَّأُ بِهِ.
- فِي مَنْطِقِ التَّرْكِيْبَةِ أَيُّ تَعَابِيرٍ رَتِيبِيَّةٍ هِيَ قَوَاعِدُ اسْتِدْلَالٍ بَدَلًا مِنْ تَعَابِيرٍ.

التحديد

التحديد هو قاعدة التخمين التي نسمح لنا بالإستباق إلى النتائج التي تظهر بأن الأشياء هي مجموعة خصائص مُؤكّدة ، p، في الحقيقة هي كُلّ الأجسام التي تملك هذه الخصائص. التحديد يُمكن أن يعالج مع استنتاج التركيبية أيضاً. لقرضاً اننا نعرف بأن : bird(tweety) اي، تويتي طير

$$\forall x: \text{penguin}(x) \rightarrow \text{bird}(x)$$

$$\forall x: \text{penguin}(x) \rightarrow \neg \text{flies}(x)$$

وان القواعد السابقة، تقول ان البطريق طير وانه لا يطير، نحن نريد إضافة الحقيقة بأنّه وبشكل نموذجي، الطيور تطير. في التحديد هذه العبارة ستكون كالتالي : أي طائر يطير إذا لم يكن شاذاً، ويُمكن تمثيل هذا حسب الآتي:

$$\forall x: \text{bird}(x) \wedge \neg \text{abnormal}(x) \rightarrow \text{flies}(x).$$

على أية حال، هذا ليس كافياً، لاننا لا نستطيع إستنتاج ان تويتي يطير بناء على السابق:

$$\text{flies}(\text{tweety})$$

حيث اننا لا نستطيع إثبات ان تويتي ليس بطائراً شاذ.

$$\neg \text{abnormal}(\text{tweety}).$$

هنا يمكن ان نطبق التحديد، وفي هذه الحالة، نحن سنفترض بأن هذه الأشياء (الاجسام) المعروضة لكي تكون شاذة، هي الأشياء الوحيدة التي ستكون شاذة. وهكذا نحن يُمكن أن نعيد كتابة قاعدتنا الأصلية كالتالي:

$$\forall x: \text{bird}(x) \wedge \neg \text{flies}(x) \rightarrow \text{abnormal}(x)$$

ورضيف التالي :

$$\forall x: \neg \text{abnormal}(x)$$

إذا أضفنا الحقيقة التالية الآن، تويتي هو بطريق:

$$\text{penguin}(\text{tweety})$$

فإنه بشكل واضح يُمكن أن نثبت بأن تويتي شاذ :

$$\text{abnormal}(\text{tweety}).$$

إذا حُددنا شاذية تويتي، نحن يمكن ان نُضيف الجملة، البطريق تويتي هو الشيء الشاذ:

$$\forall x: \text{abnormal}(x) \rightarrow \text{penguin}(x).$$

لاحظَ الإمتيازَ بين منطقِ التزكية التحديد: التزكية هي جُمَل في اللغة نفسها وليست قواعِدَ إستدلالٍ إضافية.

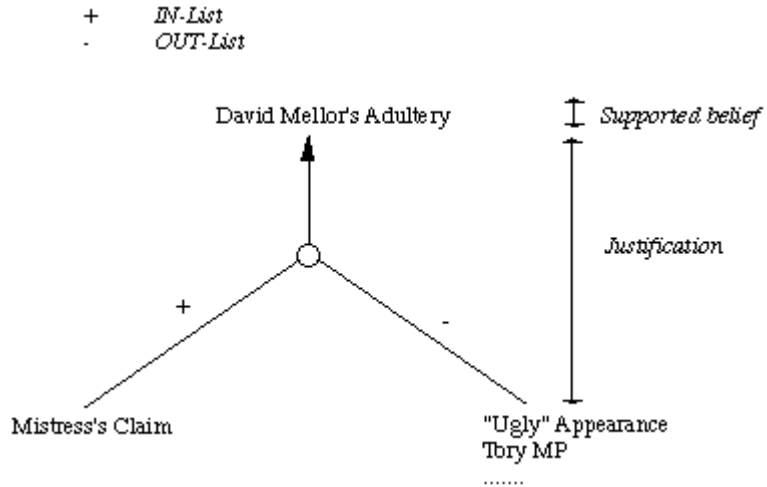
التطبيقات: حقيقة أنظمة الصيانة

العديد من أنظمة صيانة الحقيقة (TMS) طُوِّرت لوسائل تطبيق أنظمة التفكيك والاستنتاج الرتيب. مبدائياً أنظمة الصيانة لها الخصائص التالية:

- كلُّها معتمدة الى حد ما على مبدأ التراجع الموجه.
- المزام التي تحويها، مُرتبطة عن طريق شبكة التبعية.

أنظمة الصيانة المبنية على اساس التبرير (JTMS)

- هي أنظمة صيانة بسيطة لأنها لا تُعرَف أي شئ فيما يخص تركيب المزام أنفسه.
- كلُّ اعتقاد مدعوم (زعم) فيها له تبرير.
- كلُّ تبرير له جزءان:
 - في القائمة -- والذي يساند الاعتقادات المحملة.
 - خارج قائمة -- والذي يدعّم الاعتقادات التي ليست محملة.
- زعم ما مرتبط إلى تبريره بسهم.
- زعم واحد يُمكن أن يَغْذِي تبرير آخر ، هكذا يَخْلُق الشبكة.
- مزام قد تُعتبر بمنزلة اعتقاد.
- يعتبر الزعم صحيح إذا كانت كلُّ المزام في القائمة اعتقادات ولا شئ من خارج القائمة اعتقاد.
- الزعم غير رتيب اذا كان خارج القائمة ليس فارغاً أو إذا أي زعم في القائمة غير رتيب.



الوسم. JTMS 18

أنظمة صيانة حقيقة مبنية على اساس المنطق (LTMS)

هذه الانظمة مشابهة للانظمة السابقة بخلاف التالي:

- العُقد (المزام) لَنْ تَفْتَرَضَ أي علاقات بينهم ماعدا التي ستذكر بشكل واضح في التبريرات.
- JTMS يُمكن أن يُمَثَّلَ بـ P و P- بشكل متزامن. بينما LTMS يعطي تناقض هنا.
- إذا حدث هذا فيجب اعادة بناء الشبكة.

أنظمة صيانة حقيقة أساسها الفرضية (ATMS)

- "JTMS و LTMS" يُتابعان حَظَّ وحيد من التفكيك والاستنتاج في الوقت بالمتابعة التراجعية (التبعية المُوجَّهة) عند الحاجة – وذلك باستخدام بحث العمق أولاً.
- يحافظ ATMS على طرق بديلة بالتوازي – بحث العرض أولاً.
- المتابعة بالتراجع يُتفادى على حساب إبقاء السياقات المتعددة.

- على أية حال عند ظهور تناقضات الاستنتاج فإن ATMS يُمكن أن يعدل.

○ امكانية إيجاد زعم ببساطة بدون أي تبرير صحيح.

الطرق الإحصائية للتفكير المتردد

التفكير الإحصائي الرمزي

الطرق الومزية : مبدائياً تمثلُ الإعتقادات المترددة والحائِة وتصل الى احدى النتائج التالية:

- صدق، (صح)
 - خاطئ، (خطأ)
 - لا صدق ولا خاطئ.
- بعض الطرق كانَ عندها المشاكل أيضاً مع:

- المعرفة الناقصة
- التناقضات في المعرفة.

الطرق الإحصائية كما قلنا سابقاً تعطي الطريقة لتمثيل الإعتقادات التي ليست مُتأكدة (أو مجهولة) ولكن التي يُمكن لها بعض الأدلة المساندة (أو المتناقضة).

تعرضُ الطرق الإحصائية الفوائد والميزات في سيناريوه بين عريضين:

الحقيقي العشوائي

--اوراق اللعب هي مثال جيد عن هذا النوع من التفكير . نحن قد لا نستطيع توقع أي نتائج بدون أدنى شك لكننا عندنا معرفة حول إمكانية بعض العناصر (على سبيل المثال: - خصائص الآس (الرقم 1 في ورق اللعب)) ونحن يُمكن أن نستغل هذه المعرفة.

الإستثناءات

-- الطرق الومزية يُمكن أن تمثل الاستثناءات. على أية حال إذا كان عدد الإستثناءات كبيراً فإن هذا يؤدي للنظام إلى التوقف على سبيل المثال كما في العديد من مهام الحس العام والاستنتاج الخبير. بينما نجد ان التقنيات الإحصائية يُمكن أن تُلخص إستثناءات كبيرة بدون اللجوء إلى حساب.

الطرق الإحصائية الأساسية -- الاحتمال

تنبئ طرق النظرية الأساسية الإحصائية للتعامل مع الحيرة عن طريق بديهيات الاحتمال:

- احتمالات (حقيقية) أعداد في المدى 0 إلى 1.
- الاحتمال $P(A) = 0$ يُشير إلى الحيرة الكلية في $P(A) = 1$, وتأخذ قيمة بين بعض درجات الحقيقة و غير الحقيقة (الحيرة).
- احتمالات يُمكن أن تُحسب في عدد من الطرق.

ببساطة جداً

الاحتمال = (عدد النتائج المطلوبة) / (العدد الكلي من النتائج)

بحالة مثال ورق اللعب وبحال وجود مجموعة واحدة من الورق فإن احتمال وجود آساً هو 4 (عدد الآسات) / 52 (عدد اوراق اللعب في المجموعة الواحدة) و الذي يساوي 1/13. بنفس الطريقة احتمال التعامل مع ورقة بستوني هو $1/52 = 1/13$.

إذا لدينا إختيار من عدد من المواد k من مجموعة المواد n فإن الصيغة التالية تستخدم لإيجاد عدد الطرق لهذا الإختيار. (حيث، ! = عامل).

$$C_k^n = \frac{n!}{k!(n-k)!}$$

لذا الفرصة للفوز في يانصيب اللوتو (إختيار 6 من 49) هي :

$$\frac{49!}{6!43!} = 13,983,816$$

الى 1.

- الاحتمال الشرطي، $P(A|B)$ ، يُشير إلى احتمالية الحدث A , والمشروط بوقوع الحدث B .

نظرية Bayes

- تتمثل هذه النظرية بالمعادلة التالية:

$$P(H_i|E) = \frac{P(E|H_i)P(H_i)}{\sum_{k=1}^n P(E|H_k)P(H_k)}$$

- هذا يفيد بأنه إذا اعطي حدث E فإن احتمال الفرضية H_i أن تكون حقيقية يتساوى نسبة ان الاحتمال E سيكون معطاة حقيقية H_i الدليل الاستنتاجي المؤقت على احتمال H_i ومجموع احتمال E على مجموعة كل الفرضيات المؤقتة لاحتمال هذه الفرضيات.
- مجموعة كل الفرضيات يجب أن تكون متعارضة وشاملة.
- إذا نحن نفحص الدليل الطبي لتشخيص مرض ما نحن يجب أن نعرف كل الاحتمالات المسبقة للاعراض المتوفرة وأيضاً احتمال وجود مرض مستند على بعض الأعراض الملاحظة.

إحصائيات Bayesian تتمركز في قلب أكثر أنظمة التفكيك الإحصائية.

كيف يمكن ان نستغل نظرية Bayes ؟

المفتاح هو في كيفية أن نضوع المشكلة بشكل صحيح: $P(A|B)$ هي حالة احتمال A المعطى دليل B الوحيد. إذا كان هناك دليل آخر ذو علاقة فيجب أيضاً أن يؤخذ بعين الاعتبار. هنا تكمن المشكلة:

- كل الأحداث يجب أن تكون متعارضة. على أية حال أحداث المشاكل في العالم الحقيقي ليست غير مرتبطة عموماً. على سبيل المثال في تشخيص الحصبة، أعراض البقع و الحمى ذات علاقة. هذا يعني ان حساب الاحتمالات الشرطية تصبح معقدة.
- عموماً إذا الدليل الاول، p وبعض الملاحظات الجديدة، n فيمكن تنفيذ المعادلة التالية:

$$P(H|N, p) = P(H|N) \frac{P(p|N, H)}{P(p|N)}$$

نلاحظ انها تنمو تصاعدياً بالنسبة للمجموعات الكبيرة p

- كل الأحداث يجب أن تكون شاملة. هذا يعني انه لحساب كل الاحتمالات فإن مجموعة الأحداث المحتملة يجب أن تكون مغلقة. و إذا تم إنشاء معلومات جديدة فإن المجموعة يجب أن تُخلق ثانية و يجب إعادة حساب كل الاحتمالات.
- ممكن القول ان الأنظمة المبنية على اساس Bayes لاتعتبر قاعدة مناسبة للتفكيك المجهول او المتردد، لأن:
- إكتساب المعرفة صعب جداً.
- تحتاج الى الكثير من الاحتمالات – مما يؤدي الى الحاجة الى سعة تخزين كبيرة جداً.
- تحتاج الى وقت حساب كبير جداً.
- تحديث المعلومات الجديدة صعب ويستهلك وقت.
- بعض الاستثناءات لا يمكن أن تُمثل.
- البشر ليسوا مقدري احتمالات جيدين جداً.
- على أية حال، إحصائيات Bayesian ما زالت تُعتبر لب التفكير في العديد من أنظمة التفكيك المترددة، مع اضافة بعض التحسينات المناسبة للتغلب على المشاكل أعلاه.
- نحن سنناقش ثلاثة أصناف واسعة:

- عوامل الحقيقة ،
- نموذج Dempster-Shafer ،
- شبكات Bayesian.

نماذج الاعتقاد وعوامل الحقيقة

هذه النظرية اقترحت من قبل Shortliffe و Buchanan واستعملت في نظام التشخيص الطبي المشهور MYCIN.

MYCIN هو نظامٌ خبيرٌ وجوهري. هنا نُركّزُ فقط على سماتِ التفكّليّ الإحتماليّة لـ MYCIN.

- يُمثّل MYCIN المعرفةً كمجموعة قواعدٍ.
- عامل الحقيقة مرتبطٌ بكلّ قاعدةٍ.
- عامل الحقيقة مستند على إجراءات الإعتقاد B واللا إعتقاد D من فرضية H_i أعطتُ دليلاً حسب التالي:

$$B(H_i|E) = \begin{cases} 1 & \text{if } P(H_i) = 1 \\ \frac{\max[P(H_i|E), P(H_i)] - P(H_i)}{(1 - P(H_i))P(H_i|E)} & \text{otherwise} \end{cases}$$

$$D(H_i|E) = \begin{cases} 1 & \text{if } P(H_i) = 0 \\ \frac{P(H_i) - \min[P(H_i|E), P(H_i)]}{P(H_i)P(H_i|E)} & \text{otherwise} \end{cases}$$

حيث $P(H_i)$ الإحتمال القياسي.

- عامل الحقيقة C من بعض الفرضيات، معطى الدليل E والمُعَرَّفُ كالتالي:

$$C(H_i|E) = B(H_i|E) - D(H_i|E).$$

التفكير و عوامل الحقيقة

- عبرت القواعد كما لو أنّ قائمة دليل E_1, E_2, \dots فإن هناك دليل إيحائي بالإحتمال p للأعراض H_i .
- يستعمل MYCIN القواعد، للتفكير خلفياً، والوصول إلى دليل البيانات السريرية بهدف توفّر سبب مرض الكائن الحي.
- عوامل الحقيقة مجهزة مبدائياً من قبل الخبراء والتّجني يتم طبقاً للصيغ السابقة.
- كيف نؤم بللتفكّلي والاستنتاج عندما تكون عدّة قواعد مُقيّدة سوية؟ أعطتُ إجراءات الإعتقاد واللا إعتقاد عدّة ملاحظات محسوبة كالتالي:

$$B(H_i|E_1, E_2) = \begin{cases} 0 & \text{if } D(H_i|E_1, E_2) = 1 \\ B(H_i|E_1) + B(H_i|E_2)(1 - B(H_i|E_1)) & \text{otherwise} \end{cases}$$

$$D(H_i|E) = \begin{cases} 0 & \text{if } B(H_i|E_1, E_2) = 1 \\ D(H_i|E_1) + D(H_i|E_2)(1 - D(H_i|E_1)) & \text{otherwise} \end{cases}$$

- ماذا عن إعتقادنا حول عدّة فرضيات أخذتُ سوية؟ أعطتُ إجراءات ومقاييس الإعتقاد عدّة فرضيات ولَمْجها منطقياً نحسب كالتالي:

$$B(H_1 \wedge H_2|E) = \min(B(H_1|E), B(H_2|E))$$

$$B(H_1 \vee H_2|E) = \max(B(H_1|E), B(H_2|E))$$

اللا إعتقاد يحسب بنفس الطريقة.

التعلّب على عيوب قواعد Bayes

عوامل الحقيقة تلتزم بلانحة إحصائيات Bayesian، لكنّها يُمكن أن تُمثّل أنظمة المعرفة السلسلة:

- شُارك بعض القواعد الفردية الإعتقاد في الفرضيات – مبدائياً الإحتمال الشرطي.
- الصيغ لمجموعة الدليل / الفرضيات مبدائياً تفترض بأنّ كلّ القواعد مستقلة وتُستثنى الحاجة للإحتمالات المشتركة.
- إن عبء ضمان الاستقلالية يقع على عاتق كاتب القاعدة.

نماذج Dempster-Shafer

هذا يُمكن أن يُعتبر نظرة أكثر عمومية إلى تمثيل الحيرة من نظرة Bayesian. غير ان طرق Bayesian غير ملائمة أحياناً:

$$P(A) + P(\neg A) = 1$$

لنفترض ان A يُمثل الاقتراح "ديمي مور جذاب". فإن بديهيات الاحتمال تصرُّ على
تفترض الآن بأن أندرو لا يعرف حتى من هو ديمي مور. فإنه :

- نحن لا نستطيع قول ان أندرو يصدق الاقتراح، حيث انه، ليس لديه ادنى فكرة عنه.
- أيضاً، ليس من المعقول القول بأنه يُنكرُ المقترح.
- لذا سيكون من المفيد للدلالة على اعتقاد أندرو $B(A)$ و $B(\neg A)$ اعطاء الاثنين قيمة 0.
- عوامل الحقيقة لا تسمح بهذه.

حسابات Dempster-Shafer

إن الفكرة الأساسية في تمثيل الحيرة في هذا النموذج:

- حدد مجال الثقة – وهو عبارة عن مجال الاحتمالات التي يقع ضمنها الاحتمال الحقيقي مع ثقة أليّة -- مستندة على الاعتقاد B و خاصية المعقول PL المزودة ببليل E لمقترح P.
 - يجمعُ الاعتقاد كل الأدلة التي نفوذنا للاعتقاد بـ P مع بعض التأكيد.
 - تجمعُ خاصية المعقول او المقبول ظاهراً كل الأدلة المتوافقة مع P وليسيت متناقضة معه.
 - هذه الطريقة تسمح بالإضافة الأخرى إلى مجموعة المعرفة ولا تفترض نتائج مفككة.
- إذا Ω مجموعة النتائج المحتملة، فإن الاحتمال الجماعي، M، معرف لكل عضو من المجموعة 2^{Ω} وتأخذ قيم التقديرات في المدى $[0,1]$.

إن المجموعة الملعنية، \emptyset ، هي أيضاً عضو.

ملاحظة: هذا يتفاعل مع مصطلح نظرية المجموعات والذي سنعرف عليه بتفصيل اكبر في قسم لاحق.

M – هي وظيفة كثافة الاحتمال والتي تعرف َ ليس فقط من اجل Ω ولكن لـ m كل المجموعات الفرعية. إذن لو أن المجموعة Ω لها القيمة التالية :

$$\Omega = \{ Flu (F), Cold (C), Pneumonia (P) \}$$

فإن المجموعة 2^{Ω} :

$$2^{\Omega} = \{ \emptyset, \{F\}, \{C\}, \{P\}, \{F, C\}, \{F, P\}, \{C, P\}, \{F, C, P\} \}$$

- مدى الثقة معرف حسب التالي : $[B(E), PL(E)]$ حيث :

$$B(E) = \sum_A M$$

حيث أن $\Lambda \subseteq E$: كل الأدلة التي نجعلنا نعتقد بصواب P، و

$$\begin{aligned} PL(E) &= 1 - B(\neg E) \\ &= 1 - \sum_{\neg A} M \end{aligned}$$

حيث أن $\neg A \subseteq \neg E$: كل الأدلة التي تنقض P.

جَمْعُ الإعتقاداتِ

- لدينا إمكانية لتخصيص M إلى مجموعة الفرضيات.
- لدمج مصادر متعددة من الأدلة بفرضية واحدة (أو متعددة) نعمل التالي:
 M_1 و M_2 وظيفتنا إعتقاد.
- لنفترض أن X مجموعة المجموعات الفرعية Ω التي تُخصَّصُ قيم لاتساوي الصفر
ولنفترض أن Y سيُكوِّنُ نفس الشيء بالنسبة للمجموعة M_2 .
- اذن للحصول على وظيفة إعتقاد جديدة M_3 من مجموعة الإعتقادات في M_1 و M_2 نحن نستعمل المعادلة التالية:

$$M_3(Z) = \frac{\sum_{X \cap Y=Z} M_1(X)M_2(Y)}{1 - \sum_{X \cap Y=\emptyset} M_1(X)M_2(Y)}$$

بشرط $Z \neq \emptyset$

ملاحظة: نحن نعرف $M_3(\emptyset)$ لكي نكون 0 لكي يبقى الناتج المتعامد مهمة احتمال أساسية.

شبكات Bayesian

- تسمى شبكات الإعتقاد أيضاً أو شبكات الاستدلال الإحتمالية. والتي طورت بشكل مبدائي من قبل Pearl.
- إن الفكرة الأساسية قائمة على التالي:
- المعرفة في العالم قياسية -- أكثر الأحداث مستقلة بشكل مشروط عن أغلب الأحداث الأخرى.
- تبني النموذج الذي يمكن أن يستعمل تمثيل أكثر محلية للسماح للتفاعلات بين الأحداث التي تؤثر على بعضها البعض فقط.
- بعض الأحداث قد تكون فقط ذات اتجاه واحد بينما أحداث أخرى قد تكون ثنائية الاتجاه -- يُميزُ بين هذه الأحداث في النموذج.
- أحداث قد تكون سببية وهكذا تصبح مُقَيَّدة سوية في الشبكة.

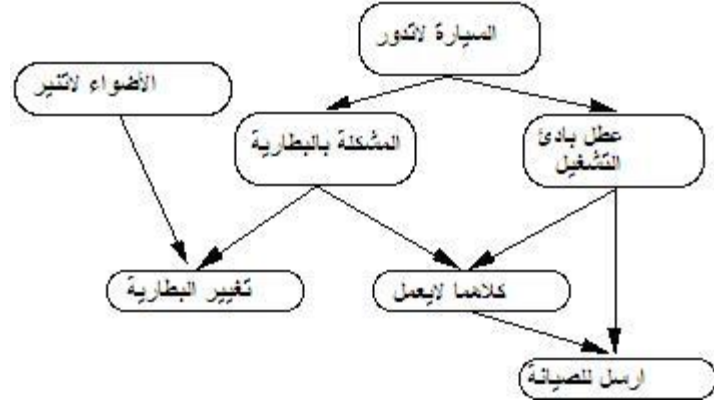
التطبيق

- أي شبكة من شبكات Bayesian هي رسم بياني (مخطط) دوري و مُقاد:
 \emptyset مخطط حيث الإتجاهات هي الصلات التي تُشيرُ إلى التبعيات التي تُجدُ بين العُقَد.
 \emptyset العُقَد تُمثلُ المقترحات حول الأحداث أو الأحداث بأنفسهم.
 \emptyset احتمالات شرطية تُحدِّدُ قوَّة التبعيات.

لنأخذ المثال التالي:

- الإحتمال، $P(\delta_1)$ بأن سيارتي لن تبدأ بالدوران.
- إذا سيارتي لن تبدأ فإنه من المحتمل ان يكون سبب ذلك:
 \emptyset البطارية تحتاج لتغيير أو
 \emptyset بادئ تشغيل السيارة عطلان.
- ليقوِّي فيما اذا كنت سأسالُح السيارة بنفسي أو أرسلها إلى المرآب للصيانة فإنني أأخذُ القرارات التالية:
- إذا الأضواء لا تعملُ فإن البطارية يُحتملُ أن تكون عطلانة لذا يمكنني تبديلها بنفسي.
- إذا بادئ تشغيل السيارة معطل فإنني اقرر إرسالُ السيارة إلى المرآب.
- إذا البطارية وبادئ تشغيل السيارة كلاهما عطلان فإننا نرسلُ السيارة إلى المرآب.

الشبكة لتمثيل هذا كالتالي:



الرسم. 19 شبكة Bayesian بسيطة

الفكر والاستنتاج في شبكات Bayesian

- احتمالات في الصلات تطبع بديهيات الاحتمال الشرطية القياسية.
- اتباع الروابط في وصول الى الفرضيات وتحديث الاعتقادات وفقاً لذلك.
- بضعة أصناف واسعة من الخوارزميات استعملت للمساعدة بهذا، مثلاً على ذلك:
 - o طريقة مرور رسائل Pearls.
 - o زمرة التقسيم الى مثلثات.
 - o طرق Stochastic.
 - o مبدائياً كل هذه الطرق تشغل العناقيد في الشبكة ويستعملون حدودهم في التأثير لتضييق البحث في الشبكة.
 - o هم أيضاً يضمنون بأن الاحتمالات محدثة بشكل صحيح.
- حيث ان المعلومات هي معلومات محلية فإنه يمكن أن تضاف و تحذف بسهولة بالتأثير الأدنى على الشبكة بأكملها. وعليه فقط العقد المتأثرة تحتاج لتحديث.

مثال عملي

- هنا نصف مثال عملي عن البحث والمستند على Cardiff.
- استعملنا شبكات Bayesian في تطبيقات نظر الحاسوب. تفاصيل العمليات البصرية المعقدة ستناقش لاحقاً في الفصل القادمة.
- هنا نحاول وصف تفكير Bayesian في كواليس العملية.
- الهدف هو تحويل المهمة المسماة بلنشاط البيانات للحصول على اقسام واجزاء - هذه الاقسام تصف الجسم (الظاهرة من مجموعة الصور) يفصل خصائصه السطحية. في المثال الماعطى هنا نحن نتعامل مع مكعب بسيط. لذا الوصف النهائي سيخرج حافاته ووجهه ولثيفية ارتباطهم سوية.
- المدخل إلى عملية الإنشطار يتكون من ثلاث مراحل والتي إنتزعت معلومات الحواف و المعلومات السطحية المستوية من صورة ثنائية الابعاد 2D بالوان تدرجات الرمادي (أحادية اللون) كذلك معلومات المدى من الصور ثلاثية الابعاد 3D.
- لذا من هذه العمليات المبدائية الثلاثة يتكون عندنا قائمة كل الخطوط، المقوسرة أو مستقيمة، قائمة كل تقاطعات الخطوط (تقاطعات خط بين او ثلاثة خطوط) و قائمة كل المعادلات السطحية التي إنتزعت من نوعي الصورة (2D,3D). يمكن الان أن نبني الشبكة من قوائم الميزات. كما هو مذكور أعلاه، نفترض تقاطع السطوح المنزوعة. لتقييم هذه الفرضيات نحتاج الى دليل لدعم الفرضية أو نقضها. إن الدليل الذي نستعمل يمكن ان يكون:
- خطوط مستقيمة إنتزعت من الصورة..
 - أقواس إنتزعت من الصورة.
 - مناطق الحيرة ' إنتزعت من خريطة العمق.

إن قائمتي الخطوط منتجتان كما وصفت فوق. يتم إيجاد مناطق الحيرة عندما تُحاول إيجاد المعادلات السطحية لكل نوع من السطوح. يتم العثور على الأخطاء في خريطة العمق حيث القناع يستخدم لإيجاد الشكل العام السطحي يتوافق مع تداخل إثنان أو أكثر من السطوح ، يميل الخطأ إلى أن يكون مكبراً لذا، يعطينا فكرة أو إشارة بأن التقاطع السطحي موجود في تلك المنطقة العامة. لذا نحن نستعمل أدلة من أكثر من مصدر واحد للبيانات. نستمر بأخذ كل من السطوح في القائمة السطحية و العقدة نولد لتمثيلها. ثم نأخذ زوج من السطوح ونحاول مقاطعتهم. إذا هم يتقاطعون فإنه من المحتمل أنها مجموعة مميزة ' والعقدة مولدة تشير الى السطوح وبنيت ببعق سطوح المستوى الأدنى (الأولاد). هذه العملية متكررة لكل زوج من السطوح التي قمنا بابتزاعها. نريد الآن أن نربط احتمال شرطي إلى كل من عقدة الجديدة. لذا نحن نعرف الآن السطوح التي من المحتمل يمكن أن تتفاعل في الجسم (الشيء). نصيف الاحتمالات إلى هذه الارتباطات. نحن نعمل هذا بإيجاد معادلة التقاطع، هذا الخط سيكون ثلاثي الأبعاد لسطحين أو بيضوي لسطح و كرة، وإسقاط هذا سطحنا البؤري. الآن عندنا تقاطعاتنا المفترضة في نفس البعد كخطوطنا المنتزعة من المرحلة التمهيدية. نقوم الآن بإيجاد ، اقرب خط مطابق من قائمة الخطوط، لكل خط تقاطع. عندما نجد الخط المماثل الأقرب نولد احتمال من الخط. لذا بوجود الخط الذي يجاري خط تقاطعنا مباشرة فإنه عندنا احتمال عالي على ان السطحان اللذان لا يتقاطعان في الجسم مستبعد تطابقهم مع الخط من قائمة الخطوط وهذا يعطينا احتمال منخفض. إن الخط الذي يتم إيجاه يُدقق أيضاً لمعرفة فيما إذا يقع ضمن منطقة الحيرة أو لا. إذا هو يقع فإن ذلك إشارة قوية أخرى بأن الخط الذي وجدناه في الحقيقة هو في موقع انضمام السطوح. لذا عندما ولدنا هذه الشبكة بكل الصلات الضرورية. فإن أكثر المعلومات التي تزود إلى النظام يمكن أن تُضاف والشبكة ستنتشر هذه المعلومات في كافة أنحاء الشبكة على شكل تحديث الاحتمال. لذا على سبيل المثال لنفترض ان صورة ما جديدة تم انشاؤها من صورة ملونة وهذه الصورة زادت إمكانية بعض الحافات والزوايا أن تكون موجودة في الصورة فإن هذا يزيد احتمال تلك الميزات التي ترتبط إلى تلك الحافات والزوايا والتي ستنتشر في كافة أنحاء الشبكة. الرسم 20 يوضح لنا مثالا بسيطاً عن الشبكة التي سولد من بيانات مساهمة الحافات والوجوه المستوية المكعب. كما يمكن رؤية، كيف ان عقد مجموعة الميزة يمكن أن تمثل المجموعات التي تتفاوت من الميزات الوحيدة مثل قطع الخط أو سطوح أو زوايا أو الجسم الكامل ممثل في العقد الأدنى التي تتضمن ثلاثة سطوح، قطع خط ثلاث، ثلاثة تقاطعات وزاوية واحدة.

المنطق الغير واضح (الضبابي)

هذا الموضوع يُعالج رسمياً أكثر في الفصول الأخرى. هنا نلخص النقاط الرئيسية فقط. المنطق الضبابي طريقة مختلفة كلياً لتمثيل الحيرة:

- يركز على حالات الغموض في وصف حيرة الأحداث بدلاً من الشك في وقوع الحدث.
- يُعزى تعاريف نظرية المجموعات والمنطق للسماح بهذا.
- تُعرف نظرية المجموعات التقليدية عضويات المجموعة كمسند منطقي.

نظرية المجموعات الضبابية

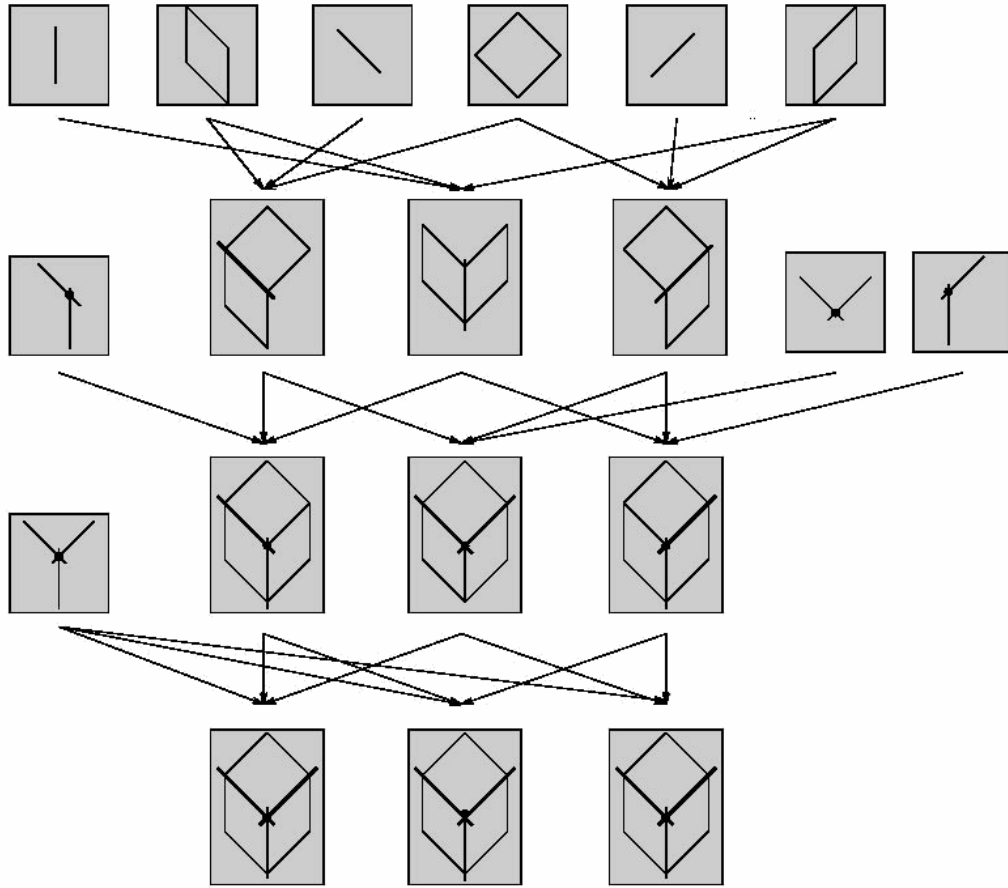
- تُعرف نظرية المجموعات الضبابية عضوية المجموعة كإمكانية التوزيع. القاعدة العامة لهذه النظرية يمكن أن تنمو كالآتي:

$$f : [0, 1]^n \rightarrow [0, 1].$$

حيث n عدد الإمكانات.

هذا يُصرح أساساً بأننا يمكن أن نأخذ n أحداث محتملة ونصل الى f كنتيجة محتملة وحيدة. هذا يمدد عضوية المجموعة حيث اننا يمكن أن نأخذ تعاريف مختلفة، لنقل ، كاري حار. شخص ما قد يعلن بأن الكاري الهندي هو الوحيد الحار جداً، بينما شخص آخر قد يقول بأن الكاري الإيراني حار أكثر. نحن يمكن أن نسمح لهذه التعاريف المختلفة للسماح لكلنا الإمكانات في التعاريف الضبابية.

- بما انه أعيد تعريف عضوية المجموعة نحن يمكن أن نطور منطق جديد مستند على دمج المجموعات. ويبرر بفاعلية.



الرسم . 20 شبكة Bayesian لإتقسام مُكعب

التفكيك الموزع

المقدمة

هناك العديد من الأسباب التي تجعلنا نريد تطبيق أو تبني نظريّة التفكير الموزع في أنظمة الذكاء الاصطناعي:

- الإنسان يستخدم نفس المبدأ -- نماذج نفسية.
 - توفر أنظمة وأجهزة تعمل بمبدأ التوازي - أجهزة وبرامج.
 - بعض التطبيقات ستُحسن الكفاءة وتُسرع التنفيذ.
- ليس كل الإجراءات قابلة للتنفيذ بشكل متوازي (تنفيذ أكثر من إجراء بنفس الوقت وبشكل متوازي). في الحقيقة الكثير من إجراءات الذكاء الاصطناعي معقدة جداً ومن الصعوبة تنفيذها بشكل متوازي. على أية حال، يُمكن كسب الكثير من السرعة على عدد العقدة لوحده. التوازي الأكثر رسمية - هو تبني أمثلة البرمجة المتوازية - وهي مفيدة جداً.
- تطوير معياري للأنظمة -- يعتبر سهل لبناء وصيانة الأنظمة وإضافة / حذف المعلومات الممثلة للمعرفة و إستراتيجيات التفكير والاستنتاج.

- تقنيات تفكيكيّ متعدّدة -- نحن يُمكنُ أَنْ نَدْعَمَ بِنَشْكِلَةِ التقنياتِ والتي يُؤدّي البعض منها المهام الاكيدة بشكل أفضل في المعرفة المُتأكّدة.
- وجهاتِ نظر متعدّدة -- المعرفة مطلوبة عندما يكون التفكيكيّ مُمثّل في عدة مجالات.
- الثقة -- إذا نموذج او جزء ما من النظام فشل بعمله فإن النظام يبقى يؤدي وظيفته ككل.

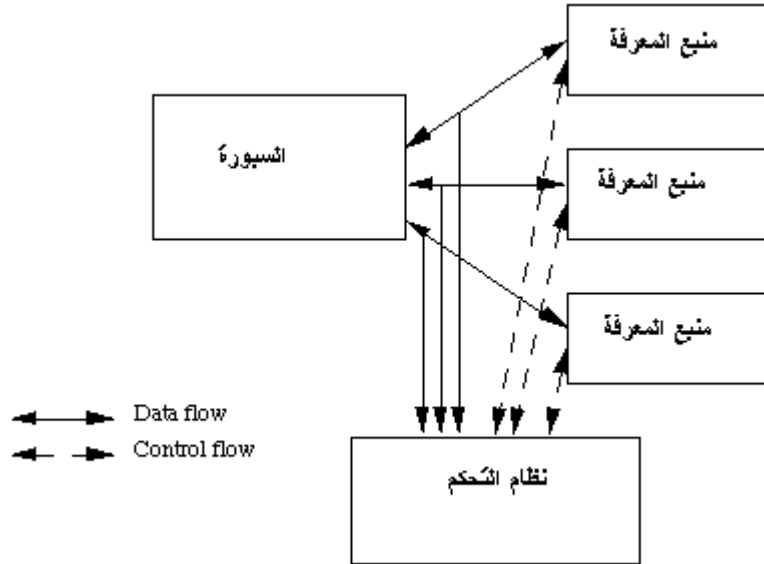
أنظمة التفكيكيّ الموزّعة

نظام التفكيكيّ الموزّع هو النظام الواحد المكوّن من وحداتٍ منفصلةٍ (تسمى العوامل) و مجموعة طرق الإتصال بينهم.

- كلّ وحدة (عامل) عادة تعمل على المشكلة التي تحلّ كياناً بنفسه.
- العديد من أنواع النظم تراوَح بين:
o أنظمة سيطرة مُتمركزة بمعرفةٍ مشتركة،
o سيطرة ومعرفة موزّعة كلياً.
- مثل هذا النظام يجبُ أَنْ يُزوّد:
- تنسيق جيد بين الوحدات.
- تراكيب إتصال جيدة.
- تقنيات تفكيكيّ موزّعة.
- اذن على الانظمة ان نتعامل مع سمات التنسيق العام، و تراكيب الإتصال والسيطرة. في هذا الفصل نحن سنناقش تركيبين معينين والذان أصبَحَا آليات التفكيكيّ الموزّعة الأكثر شهرة:
- أنظمة اللوح (السيبورة).
- أنظمة عبور الوسالة.

أنظمة السبورة

إنّ الفكرة الأساسية من نظام السبورة موضحة في الرسم 21.



الرسم 21 نموذج السبورة

النظام يتكوّن من:

مصادر المعرفة

-- مجموعة الوحدات المستقلة التي تحتوي على المعرفة في المجالات المعيّنة.

السبورة

-- بنية البيانات المشتركة والتي من خلالها تتصل مصادر المعرفة المختلفة.

نظام السيطرة

-- بقوَر في امر أي مصادر المعرفة سَتَشْتَغَلُ على المداخل في السبورة.

استنتاج وتنظيم المعرفة

- تركيب وبنية السبورة إلى حد كبير منظم.
- نظام السبورة يَصِفُ تنظيم مجالات المعرفة وكُلّ المداخلات، كذلك الحلول المتوسطة والجزئية المطلوبة لحلّ المشاكل.
- مجال الحلّ منظم إلى واحد أو أكثر من التطبيقات المعتمدة على الهيكلية الهرمية.
- المعلومات في كُلّ مستوى من الهرم نُقْلُ حلول جزئية.
- مجالات المعرفة مُقسّمة إلى مصادر المعرفة الفردية.
- تنجز التحويلات في هيكلية السبورة باستعمال إجراءات إرشادية أو حسابية باستِجَاب وحداث المعرفة تحت نظام السيطرة.

لنفترض ان مجموعة من الناس تُجمَعُ لعبة الصور المقطعة مع سبورة:

- يُنْظَر المتطوعون إلى القطع ويقومون بأختيار القطع الأكثر ملائمة ووضعها على اللوحة.
- كُلّ عضو يُنْظَر على قطعه ويرى إذا كانت متوافقة على اللوحة. بمثل هذه القطع نصل الى الحلّ.
- الحلول الجديدة تؤدي الى وضع قطع أخرى في مكانها.
- ليكون الحلّ حلّ سبورة يجب التالي:
- لا إتصال مباشر يُسمَحُ بوقوع الحدث. لعبة الصور المقطعة يُمكنُ أَنْ تُجمَع بصمتٍ كلي و بدون مشاكل حقيقية.
- لا يَهَم مَنْ وضع اي قطع.
- كُلّ الإتصالات بصري على اللوحة.
- كُلّ شخص عنده تَنَشِيط ذاتي -- يعرف عندما تكون القطع التي يضعها مطابقة.
- لا طلب إستنتاجي مؤسّس – يعني تعاون مُتَوَسَّط بحالة الحلّ على السبورة.
- الحلّ يُجمَعُ بشكل تزايدى وبانتهازية – وهذه هي الخواص المهمة في أنظمة الذكاء الاصطناعي المبنية على اساس السبورة.

لإيضاح أهمية وحدة السيطرة لنمعن النظر في المشكلة أعلاه:

- لنفترض ان السبورة تقع في ممر ضيق، بحيث فقط شخص واحد يُمكنُ أَنْ يُنْظَر للوحة في حالة وحيدة.
- نحتاج الى مراقب للسيطرة على مراقبة اللوحة وعملية تَجَمُّع الصورة المقطعة.
- هذه العملية يُمكنُ أَنْ تُكوّن بسيطة جداً أو يُمكنُ أَنْ تُتَصَمَّن الكثير من الإستراتيجيات المعقّدة، مثل إظهار قطع كُلّ شخص والتقرير بشأن القطع التي تلائم `` بشكل أفضل ``.
- العناية مطلوبة لكي لا يُهاوَم على الخاصية الضرورية الواحدة لنظام السبورة، والتي تعتبر حلّ المشكلة الإنتهازي.
- التطبيقات المبنية على اساس السبورة تواجه مشاكل مماثلة في أغلب الأحيان.

قضايا تطبيق

يُصَوِّر المثال أعلاه بعض المشاكل التي قد تظهر عند تُطَبِّقُ نظام السبورة. في هذا القسم نأخذ بعين الاعتبار بعض الاعتبارات العامة المطلوبة عند تطبيق وتصميم إطار السبورة.

مصادر المعرفة

- مجالات المعرفة يَجِبُ أَنْ تُقسَمَ لكي تبقى المصادر منفصلة ومستقلة.
- كُلّ مصدر معرفة يجب ان يكون مسؤولاً عن معرفة الشروط التي هو تحتها يُمكنُ أَنْ يُساهم حلّ:
 - مجموعة الشروط المبدائية تُحدّد دلائل القبول قبل تفعيل المصدر.
- مصادر المعرفة مُمَثَّلَة كمجموعات القواعد، لمزاعم المنطق أو الإجراءات.

- مصادر المعرفة تعدل السبورة فقط أو تُسيطر على تراكيب البيانات. و فقط مصادر المعرفة هي الوحيدة التي يمكن ان تعدل السبورة (الرسم 21).

تراكيب بيانات السبورة

- مصادر المعرفة تحدد التغييرات للتركيب الذي يفود بشكل تزايدى إلى الحل.
- تراكيب البيانات تشمل الأجسام من حيز الحل:
 - o البيانات المدخلة،
 - o الحلول الجزئية،
 - o الهدائل،
 - o الحلول النهائية،
 - o بيانات السيطرة والتحكم.
- الشكل النهائي للتركيب هو مشكلة اعتماد، بالرغم من أن الأجسام تُنظم بشكل هيكلي هرمي يعني تراكيب من نوع : شجرة / رسم بياني.
- مراتب متعددة أو طبقات سبورة ممكنة هنا.

السيطرة والتحكم

- مصادر المعرفة تتجاوب بانتهازية إلى التغييرات على السبورة.
- مجموعة وحدات السيطرة تُراقب التغييرات على السبورة و a الجدول يُقرّر اي الأعمال يجب أن يتّخذ في الخطوة التالية.
- أنواع مختلفة من المعلومات تجعل متوفرة بشكل عام لوحدة السيطرة ومخزونة أما :
 - o على السبورة، أو
 - o بشكل منفصل في وحدة التحكم.
- الهدف منهم هو نتائج الإنتباه.
 - بؤرة الإنتباه يُمكن أن تكون:
 - o تراكيب السبورة ، أو
 - o مصادر المعرفة ، أو
 - o كلاهما.
- الحلول عُرّزت مرحليا مع الوقت. العديد من أنواع التفك يّ طبقت (ومثال على ذلك: - توجيه البيانات ، توجيه الهدف، توجيه التوقعات).

حل واستنتاج المشكلة

- تحدث الخطوات التكرارية التالية ضمن نشاطات حل السبورة المثالية:
1. مصدر المعرفة يقوم بالتغيير على اجسام السبورة.
 2. كل مصدر معرفة يُشير إلى المساهمة الذي هو يُمكن أن ي قدمها للوصول إلى حالة الحل الجديد -- أما بالاستناد على تبادل البيانات ديناميكيا أو ببساطة.
 3. الجدول يُقرّر بؤرة الإنتباه.

3

1. وحدة السيطرة تُهيئ بؤرة الإنتباه للتنفيذ:

1. إذا بؤرة الإنتباه كانت مصدر المعرفة – فإن اجسام السبورة تُختار لتخدم كسياقها (جدولة المعرفة المركزية).
2. إذا بؤرة الإنتباه كانت جسم من اجسام السبورة -- فإن مصدر المعرفة لمعالجة ذلك الجسم مختار ومعطى مع الجسم بشكل سياقه (جدولة الاحداث المركزية).
3. إذا كان كل من اجسام السبورة و مصدر المعرفة هما بؤرة الإنتباه لحالة مصدر المعرفة والتي تكون جاهزة للتنفيذ مع الجسم كسياقها.

1. عملية حل المشكلة تستمر حتى يشير مصدر المعرفة بأنها يجب أن تتوقف وذلك حسب التالي:

1. أما بالحصول على حل مقبول ، أو
2. تمنع قلة المعرفة أو البيانات عملاً آخرًا.

امثلة على هذا النوع من الانظمة

نعرض هنا بعض انواع الانظمة العملية المبنية على اساس السبورة:

HEARSAY II

-- السبورة الأصلية: نظام تمييز الصوت.

HEARSAY III

-- نسخة مُعَمَّمة من النظام السابق.

BB1,GBB

-- سبورات متعددة الاغراض.

HASP

-- مراقبة إشارة السونار (جهاز لكشف وجود الاشياء تحت الماء بواسطة موجات صوتية)

TRICERO

-- سبورة موزعة تُستعمل لمراقبة المجال الجوي.

انظمة عبور الرسالة

هو عبارة عن نظام اتصالات بديل للعملاء.

انظمة عبور الرسالة تختلف عن انظمة السبورات في الطرق التالية:

- تقبل العوامل إلى معرفة أكثر حول بعضهم البعض.
- المعرفة تُستعمل لتوجيه الرسائل، التي على الأغلب، لتنفيذ المهمة المطلوبة.
- يمتلك كل عامل المعرفة الكافية بحيث يمكن استخدام هذه الخبرة بدون الإتصال مع الوكلاء الآخرين.
- تعالج المعلومات ثم تمر للتالي.

نظم عبور الرسالة المثالية تمتلك المكونات التالية:

وكلاء حل المشكلة

-- خبراء في مجال معين.

وكلاء نظام

-- مفسرات الاوامر ، معالجة الخطأ... الخ.

الوسائل

-- الوظائف الضمنية في النظام والتي يمكن أن يستعملها الوكلاء لمطابقة النمط.

قاعدة بيانات وصف الوكيل

-- محدث في النظام.

النواة

-- واحد لكل معالج لمعالجة الإدخال / الإخراج، توجيه الوسطاء... الخ.

وكيل حل المشكلة سيبقي نماذج الوكلاء الآخرين. هو سيُشكّل المعلومات التالية نموذجياً حول وكيل ما:

- الاسم، الوصف والموقع (العنوان).
- العلاقة أو الدور بين الوكيلين.
- مهام و مهارات الوكيل.
- الأهداف أو نتيجة (مخرجات) الوكيل
- خطة الوكيل -- كيف يُنجز أهدافه.

معالجة اللغة الطبيعية

المقدمة

في هذا الفصل سنناقش واحدة من أهم المهام الحياتية، وهي اللغة، وليبيان هذه الأهمية، نحن سنقارنها في بعض الأحيان مع الرؤية "النظر". اللغة والرؤية الطبيعية. كلتا المهمتين متعلقة أولاً بالفهم: في اللغة الطبيعية نحن نتعلق بفهم اللغة المنطوقة أو المطبوعة، بينما في الرؤية نحن مهتمون بفهم الصور. في كلتا الحالتين نحن نحاول الحصول، من بعض بيانات المدخل الحسي، على بعض التمثيل حول المعنى الحقيقي لتلك البيانات. بالطبع "المعنى الحقيقي لتلك البيانات" هي مبهمة بعض الشيء - بحيث أننا لسنا مهتمون عموماً بالنتائج الفلسفية العميقة للبيانات (ومثال على ذلك: -، شروق الشمس الجميل يعني أن هناك إحساس نحو الكون...). نحن فقط مهتمون بالتفسير الكافي لأهدافنا. في الرؤية ربما نحن فقط نريد التعرف على قطع متقلقلة من المشهد ككل لذا نظام البصر الآلي يمكن أن يزيلهم من المشهد - "معنى" الصورة يتعلق بتصنيف الصور. في اللغة الطبيعية نحن نريد أن تكون قادرة على إجابة أسئلة مستخدمها بإعطاء بعض المعلومات من قاعدة البيانات. عليه ممكن القول أن معنى الجملة قد يشبه استفسار معطى لقاعدة بيانات. للمهام الأخرى، تمثيل المعنى قد يكون مختلف جداً، ومهمة ال فهم تتراوح بين سهلة وصعبة حسب المدخل.

المشكلة

أنظمة اللغة الطبيعية طور ت لإستكشاف النظريات اللغوية العامة ولإنتاج وصلات اللغة الطبيعية أو الواجبات الأمامية للتطبيقات البرمجية. سنفترض بأن هناك بعض التطبيقات عموماً فلق المستخدم يتفاعل مع ها، وبأن مهمة النظم الذكية هي تفسير، تعابير المستخدم وترجمتها إلى الشكل المناسب للتطبيق. نحن سنفترض أيضاً الآن أن اللغة الطبيعية موضع السؤال ممكن أن تكون أي لغة حية كالعربية أو الانكليزية... الخ. عموماً المستعمل قد يتصل بالنظام ب الكلام المنطوق أو بالطباعة. فهم اللغة المنطوقة اصعب بكثير من فهم اللغة المطبوعة.

سواء نحن بدأنا بإشارات الخطاب (ادوات تمثيل اللغة المنطوقة) أو مدخلات الطباعة (اللغة المطبوعة)، في مرحلة ما سيكون عندنا قائمة (أو قوائم) من الكلمات ويجب أن نعمل لمعرفة معانيها. هناك ثلاث مراحل رئيسية لهذا التحليل:

- التحليل النحوي: حيث نستعمل قوانين قواعدية تصف التركيب القانوني للغة للحصول على اعراب واحد أو أكثر للجملة.
- التحليل اللفظي: حيث نحاول ونحصل على تمثيل أولي (أو تمثيلات) لمعنى الجملة.
- التحليل الواقعي: حيث نستعمل معلومات سياقية إضافية لملى الفجوات في تمثيل المعنى، ول معرفة ما اراد المتكلم ايصاله من معنى.

النحو

إن مرحلة التحليل النحوي أفضل مرحلة مفهومة لمعالجة اللغة الطبيعية. النحو يساعدنا على فهم كيفية جمع الكلمات لتشكيل جمل معقدة، ويطينا نقطة بداية لفهم معنى الجملة الكاملة. على سبيل المثال، لنفترض الجملتين التاليتين:

- أكل الكلب العظم.
 - العظم أكل من قبل الكلب.
- قواعد النحو تساعدنا لفهم بأن العظم هو الذي أكل وليس الكلب. يسمح التحليل النحوي لنا لتقرير تجمعات محتملة من الكلمات في جملة. أحياناً هناك فقط سيكون تجميعاً محتملاً واحد، ونحن سنكون في الطريق الصحيح لفهم المعنى. على سبيل المثال، في الجملة التالية:

- الأرنب ذو الأذنان الطويلة يستمتع بالخس الأخضر الكبير.
- نحن يمكن أن نستنتج من قواعد النحو بأن "الأرنب ذو الأذنان الطويلة" تشكل مجموعة واحدة (جملة إسمية)، و "الخس الأخضر الكبير" تشكل مجموعة جملة إسمية أخرى. عندما نريد فهم معنى الجملة علينا أن نبدأ بفهم معنى كلمات هذه المجموعات، قبل دمجهن سوياً للحصول على معنى الجملة الكاملة.

في الحالات الأخرى قد يكون هناك العديد من التجمعات المحتملة للكلمات. على سبيل المثال، الجملة "رأى زيد عمر بالنظارات" هناك قراءتان مختلفتان مستندة على التجمعات التالية:

- زيد رأى (عمر بالنظارات). تعني ان ، عمر يضع نظارات.
 - زيد (رأى عمر بالنظارات). تعني ان ، زيد يضع النظارات و رأى عمر من خلالها.
- عندما يكون هناك العديد من التجمعات المحتملة فان الجملة تكون غامضة بشكل نحوي. أحياناً نحن سنكون قادرين على إستعمال المعرفة العامة لفهم التجميع المقصود - على سبيل المثال، الجملة التالية:
- رأيتُ برج إيفل طائراً الى باريس.
- نحن يُمكنُ أنْ نُحْزِرَ أنَّ البرج لا يَطِيرُ! لذا، هذه الجملة بشكل نحوي غامض، لكن المعنى الصحيح انني رأيت البرج بينما انا طائراً الى باريس.
- على أية حال، نُحدِّدُ قوانين النحو التنظيم المحتمل للكلمات في الجُمْلِ. القوانين تحدد عادة بكتابة قواعد اللغة. بالطبع، وجود القواعد وحدها غير كاف لتحليل الجملة - نحتاجُ الى مُعْرَب لُغوي لإستعمال القواعد لتحليل الجملة. المُعْرَب اللغوي يَجِبُ أنْ يُرْجَعَ شجرات الإعراب المحتملة للجملة، يُشيرُ إلى التجمعات المحتملة للكلمات. القسم القادم سيَصِفُ كَيْفَ يمكن ان نكتب قواعد بسيطة ومُعْرَبات لُغوية، بالتركيز على بناء لغة البرمجة Prolog ، في شكلية قواعد البندِ المباشرة.

كتابة القواعد

قواعد اللغة الطبيعية تُحدِّدُ بنية جملة جائزة من ناحية الأصناف النحوية الأساسية مثل الأسماء والأفعال... الخ، وتسمح لنا لتقرير تركيب الجملة. هو مُعْرَفٌ عَلَى نحو مماثل بالنسبة إلى قواعد اللغة برمجة ما، مع ذلك يَمِيلُ إلى أن يكون أكثر تعقيداً، ومجموعة الرموز المستعملة مختلفاً جداً. بسبب تعقيد اللغة الطبيعية، فان القواعد المعطاة لا يبدو أنها تُعْطِي كُلَّ الجُمْلِ المحتملة و المقبولة بشكل نحوي.

[ملاحظة: في اللغة الطبيعية نحن لا نُعْرَبُ اللغة عادة لتُدَقِّقَ بأنّها صحيحة. ولكن نُعْرِبُها لكي نحدد التركيب ونُساعدُ على فهم المعنى.

نقطة البداية لوصف تركيب لغة طبيعية هو أن نَسْتَعْمَلَ سياق قواعد حرة (كما غالباً نُسْتَعْمَلُ لوصف النحو في لغات البرمجة). لِنَقْرَضُ اننا نريدُ القواعد التي ستُعْرَبُ جُمْلَ مثل:

- أكل زيد البسكويت.
- أكل الأسد المريض بالفصام.
- قَبْلَ الأسد زيد.
- لكننا نريدُ إستثناء جُمْلَ خاطئة مثل:
- أكل البسكويت جون.
- المريض بالفصام أكل الأسد.
- قبل البسكويت الأسد.

القواعد البسيطة التي تتعامل مع هذا هي التالية:

- 1. الجملة --> عبارة _ اسمية، عبارة _ فعلية.
- 2. الجملة الاسمية --> اسم _ صحيح.
- 3. الجملة _ الاسمية --> أداة التعريف، اسم.
- 4. الجملة _ الفعلية --> فعل، اسم _ عبارة.

الاسم _ الصحيح --> [زيد].

الاسم _ الصحيح --> [عمر].

اسم --> [مريض بالفصام].

اسم --> [بسكويت].

فعل --> [أكل].

فعل --> [قَبْل].

أداة تعريف --> [].

إنَّ الترميز مشابه للذي يهتم عمل أحياناً لقواعد لغات البرمجة. الجملة تشتمل على عبارة اسمية أو/ و عبارة فعلية. العبارة الإسمية تشتمل أما على اسم علم (استعمال قاعدة 2) أو أداة التعريف (ومثال على ذلك: -، ال) و اسم. العبارة الفعلية تشتمل على فعل (ومثال على ذلك: -، أكل) و عبارة إسمية. إنَّ القواعد في النهاية حقاً مثل مداخل قاموس، التي تُذكرُ الصنفَ النحوي للكلمات المختلفة. الأصناف النحوية الأساسية مثل `` اسم " و `` فعل " رموزاً طرفية في القواعد، كما هم لا يُمكن أن يُوسَّعوا إلى أصناف المستوى الأدنى. (وَضَعْنَا الكلمات في الأقواس المربعة لأن ها هذه هي الطريقة التي يُعملُ بها عموماً في Prolog).

للقواعد المُعطاة والتي نحن يُمكن أن نُصوِّر التركيب النحوي للجملة بإعطاء شجرة الإعراب، والتي تظهر كيف ان الجملة تُقسَّم إلى الاجزاء النحويين المختلفين. هذا النوع من المعلومات قد يكون مفيداً للمعالجة اللفظية.

بالطبع، القواعد المعطاة فوق ليست كافية جداً لإعراب لغة طبيعية بشكل صحيح. إعتبر الجملتين التاليتين:

- أكلُ زيد الأسد.
 - يأكلُ زيد الأسد الشرس.
- إذا عُدْنَا `` أكل " و `` يأكل " صنفَت كأفعال ، فإذا اخذنا القواعد البسيطة الواردة اعلاه ، فإن الجملة الأولى سَتَكُونُ مقبولة طبقاً للقواعد، بينما الثانية لا – حيث انه لا يوجد عِنْدْنَا أي ذِكْر صفات في قواعِدنا. لحل هذه المشكلة نَحْتَاجُ لإضافة قواعد (قوانين) أخرى إلى قواعِدنا.

اسم _ عبارة (ضمير) --> مصمم (ضمير)، صفات، اسم (ضمير).
 صفات --> صفة، صفات.
 صفات --> صفة.

صفة --> [شرس].

صفة --> [قبيح].

الخ.

الشيء الآخر الذي من الضروري أن نعمله لقواعدنا هي ان نُوسِّعها لذا نحن يُمكن أن نُميز بين الأفعال المتعدية التي تأخذ جسماً (ومثال على ذلك: -، يحب) وأفعال لازمة التي لا تحتاج الى جسم (ومثال على ذلك: -، يتحدث). (`` يَحُبُّ عمر الأسد " صحيحة، بينما `` يُناقشُ عمر الأسد " ليست صحيحة منطقياً).
 قواعِدنا حتى الآن ما زالت تُعَرِّبُ جُمَلٌ فقط بشكل بسيط جداً. يُمكن أن نُحاول إضافة قواعد أكثر فأكثر - على سبيل المثال، نَحْتَاجُ القواعد التي تتعامل مع عبارات الجرّ ، و الاسماء الموصولة. الخ.

القواعد في لغة البرمجة Prolog

إنَّ الشكليّة العامّة التي إستعملنا فوق مستندة على بنية شكلية القواعد في Prolog ، المعروفة بقواعد الفقرة المباشرة. يَسْمَحُ Prolog بكتابة القواعد التي نشتغل مباشرة مجموعة الرموز (كتلك الواردة في المثال السابق)، لكن داخلياً يتم ترجمتها إلى قواعد Prolog عادية. مثل:
 جملة --> اسم _ عبارة، فعل _ عبارة.

sentence --> noun_phrase, verb_phrase.

نُترجم داخلياً إلى الشكّل مثل:

```
sentence(Words, Remainder) :-
    noun_phrase(Words, NPRemainder),
    verb_phrase(NPRemainder, Remainder).
```

المُعرِّبات اللُّغوية

إملاك القواعد غير كاف لإعراب لغة طبيعية - نحن نحتاج إلى مُعرب لغوي. المُعرب اللغوي يجب أن يبحث عن الطرق المحتملة بحيث يمكن استعمال القواعد لإعراب الجملة - لذا الإعراب يُمكن أن يعتبر كنوع من مشكلة بحث. عموماً هناك قد يكون العديد من القواعد المختلفة التي يمكن أن تُستعمل لتوسيع أو إعادة كتابة الصنف النحوي المعطى، والمُعرب اللغوي يجب أن يفحصهم كل هم، لمعرفة إذا الجملة يُمكن أن تُعرب أو لا. على سبيل المثال، في قواعدنا الصغيرة بالمثال اعلاه، كان هناك قاعدتان للاسم - عبارات: إعراب الجملة قد يستعمل أما واحد أو الآخرون. في الحقيقة نحن يُمكن أن نمثل القواعد كتعريف شجرة AND-OR للبحث.

لذا، لإعراب جملة نحن نحتاج للبحث خلال كل هذه الإمكانيات، نمرُّ بكلِّ التراكيب النحوية المحتملة عملياً لإيجاد تركيب واحد يُلائم الجملة. هناك طرق جيدة وطرق سيئة لتنفيذ هذا، كما هناك طرق جيدة وسيئة لإعراب لغات البرمجة. طريق واحد أساسي هو البحث بالعمق أولاً خلال شجرة الإعراب. عندما نصل العقدة الطرفية الأولى في القواعد (الصنف النحوي الهادي، مثل الاسم) ندقق الكلمة الأولى للجملة إذا ما تعود إلى هذا الصنف (ومثال على ذلك: -، اسم). إذا هـ ي الكلمة المطلوبة، فإننا نواصل إعراب بقية الجملة. إذا لم يكن هو المطلوب نراجع ونحاول قواعد القواعد البديلة.

كمثال، لِنَقْرَضُ بِأَنَّكَ كُنْتَ تُحَاوِلُ إِعْرَابَ ``يَحِبُّ زَيْدٌ عَمْرُ`` أعطى القواعد التالية:

جملة --> اسم - عبارة، فعل - عبارة.

فعل - عبارة --> فعل، اسم - عبارة.

اسم - أداة تعريف --> -، اسم - عبارة

اسم - عبارة --> P اسم.

فعل --> [يحبُّ].

p اسم --> [زيد].

p اسم --> [عمر].

sentence --> noun_phrase, verb_phrase.

verb_phrase --> verb, noun_phrase.

noun_phrase --> det, noun.

noun_phrase --> p_name.

verb --> [loves].

p_name --> [john].

p_name --> [mary].

أنت قد تبدأ التوسع لـ جملة " إلى عبارة فعلية و عبارة اسمية. ثم العبارة الاسمية ستوسع لإعطاء أداة تعريف و اسم، يستعمل القاعدة الثالثة. أداة التعريف صنف نحوي (عقدة طرفية في القواعد) لذا ندقق إذا ما الكلمة الأولى (زيد) تعود إلى ذلك الصنف. الجواب لا - زيد اسم علم - لذا نترجع ونجد طريق آخر من التوسع ``اسم - عبارة" ونحاول القاعدة الرابعة. الآن، كزيد اسم علم هذا سيعمل بشكل حسن، لذا نواصل إعراب بقية الجملة ``يحبُّ عمر". نحن لحد الآن لم نوسع عبارة فعل، لذا نحاول إعراب ``يحبُّ عمر" كعبارة فعل. هذا سينجح في النهاية، لذا الامر بأكمله ينجح.

قد يكون واضح حتى الآن ان الية الاعراب في Prolog، يرتبط في إجراء البحث فقط. Prolog سيتراجع فقط لإستكشاف التراكيب النحوية المحتملة المختلفة. الحجج الإضافية ال موجودة في Prolog (داخلياً) ستسمح له بقراب من الجملة باستعمال قاعدة، ثم بقية الجملة باستعمال قواعد أخرى.

لاحظ بأن هذا نوع بسيط من انواع الاعراب " من الأعلى للأسفل". مُعرب لغوي يُمكن أن يُطبق إلى حد معقول بسهولة في اللغات الأخرى، التي لا نضعم التراجع، باستعمال آلية بحث أساسها جدول الأعمال.

المُعربات اللغوية البسيطة غير كفاء في أغلب الأحيان، لأنهم لا يبقون سجل كل قطع الجملة التي أعربت. استعمال البحث البسيط - العمق أولاً، بللترجع يُمكن أن يؤدي إلى القطع المفيدة من الإعراب.

مُعرب شبكة الإنتقال اللغوي يَعْبُرُ هذه الشبكة ، ويُدَقِّقُ فيما إذا كانت الكلمات في الجملة تجاري الأصناف النحوية على كُلِّ قوسٍ أو لا .

دراسة معاني الكلمات و فلسفة الذرائع

المرحلتين الباقيتين من التحليل، وهما دراسة معاني الكلمات فلسفة الذرائع، مُهتَمَتَيْن بِلِلتَوَصُّلِ إلى معنى الجملة. في المرحلة الأولى (دراسة معاني الكلمات) نحصل على التمثيل الجزئي من المعنى بالإستزاد على التركيب / التراكيب النحوي المحتمل للجملة، وعلى معاني الكلمات في تلك الجملة. في المرحلة الثانية، يوضح المعنى بالإستزاد على المعرفة السياقية (في سياق الكلام) والعامة. لتصوير الاختلاف بين هذه المراحل، لنأخذ الجملة التالية :

- سأل عن الرئيس.

من معرفة معنى الكلمات وتركيب الجملة الذي نحن يُمكنُ أَنْ نَحْسَبَ بأنَّ شخص ما (ضمير مستتر تقديره هو) سأل عن شخص ما الذي هو رئيس. لكننا لا نستطيع أن نقول مَنْ هؤلاء الناس ولماذا الرجل الأول يريد الثاني. إذا عَرَفْنَا شيء حول السياق (من ضمن ذلك الجُمْلِ الأخيرة بَصْنَعِ جمل محكية / مكتوبة) نحن قَدْ نُكُونُ قادرون على حَلِّ هذه الأشياء. لَرُبَّمَا الجملة الأخيرة كانت " فريد طرد من عمله. "، ونحن نَعْرِفُ من خلال معرفتنا العامة رؤساء العمل يمكن أن ينهوا خدمات عامل ما. وإذا يُرِيدُ شخص ما أَنْ يَتَكَلَّمَ مع الشخص الذي فَصَّلَهُ من عمله فإنه للإعتراض عليه. نحن يُمكنُ أَنْ نَبْدَأُ بِلِلتَوَصُّلِ إلى معنى الجملة حقاً - فريد يُرِيدُ القَظْمِ إلى رئيسه حول فصله من العمل. على أية حال، هذه المرحلة الثانية لِلتَوَصُّلِ إلى المعنى السياقي الحقيقي مدعوة باسم فلسفة الذرائع. المرحلة الأولى - مستندة على معاني الكلمات وتركيب الجملة - دراسة معاني الكلمات وهذا ما نحن سَنناقشُ بشكل مفصل أكثر.

دراسة معاني الكلمات

عموماً، المدخل إلى المرحلة اللفظية للتحليل قَدْ يُظَرُّ كما أن لو تَكُونُ مجموعة محتملة لإعراب الجملة، ومعلومات حول معاني الكلمة المحتملة. إنَّ الهدفُ أَنْ يَدْمَجَ معاني الكلمة، المعرفة المعطاة لتركيب الجملة، للحصول على تمثيل أولي من معنى الجملة الكاملة. إنَّ الشيء الصعب، بمعنى، أَنْ يُمَثِّلَ معاني الكلمة بطريقة بحيث أَنَّهُمْ قَدْ يندمجون مع معاني الكلمة الأخرى بطريقة بسيطة وعامة. لنأخذ الجمل التالية:

- يُطَيَّرُ الوقت مثل السهم.

- تطير الفياكه مثل الهوزة.

إذا عِنْدْنَا بَعْضَ تمثيل معاني الكلمات المختلفة في الجملة فإنه يُمكنُ أَنْ نَسْتَنِي الأعراب السخيف المحتمل. بالظُّرِّ إلى المثل في الأعلى وتحديدًا إلى كلمة " موز " ونحن نعرف بأنَّه فاكهة، والشار عموماً لا نَظَيِّرُ. إذن نحن نستنتي القراءة " نَظَيِّرُ الفواكه مثل موزة ". إذا تَأَكَّدْنَا بأنَّ الجُمْلِ التي تُعْنِي " X تَعْمَلُ شيءًا مثل 'Y' فإن ذلك يَتَطَلَّبُ بأنَّ X و Y يُمكنُ أَنْ يَعْمَلَ ذلك الشيء!

أحياناً يكون الغموض مُقَدَّمُ في مرحلة التحليل اللفظي، على سبيل المثال:

- دَهَبَ عمر إلى المصرف.

هل دَهَبَ جون إلى مصرف النهر أو المصرف المالي؟ نحن نُريدُ أَنْ نَجْعَلَ هذا واضحاً في تمثيلنا السيمانطيقي، لكن بدون معرفة سياقية نحن لَنَيسَ عندنا طريقة جيدة للاختيار بينهم. يَحْدُثُ هذا النوع من الغموض عندما كلمة ما لها معنيان محتملان، لكن كلاهما ، على سبيل المثال، قد يَكُونُ أسماً.

للحصول على تمثيل لفظي يُساعدُ إذا نحن يُمكنُ أَنْ نَدْمَجَ معاني أجزاء الجملة بطريقة بسيطة لِلتَوَصُّلِ إلى معنى كُلِّي (تُشِيرُ دراسة معاني الكلمات التعبير التركيبية إلى هذه العملية). لتلك المألوفة بتعابير λ ، طريق واحد لَعْمَلُ هذا وذلك بَلِّغُ يُمَثِّلُ معاني الكلمة كتعابير λ معقدة، فقط إستعمال تطبيقاً وظيفياً لدمجهم.

فلسفة الذرائع

فلسفة الذرائع هي المرحلة الأخيرة للتحليل، حيث أنَّ المعنى يوضح بتفصيل أكبر بالإستزاد على المعرفة السياقية والعامة. تَتَضَمَّنُ المعرفة السياقية معرفة الجُمْلِ السابقة (منطوقة أو مكتوبة)، معرفة عامة حول العالم، ومعرفة المتكلم.

وظيفة مهمة واحدة في هذه المرحلة وهي أن نحدد الضمائر الخاصة بالتعابير. على سبيل المثال، في الجملة `` هو اطعم القط الابيض " التعبير `` القط الابيض" يشير إلى قط ابيض معين (مثال ، توم). الضمير `` هو " يشير إلى الرجل المعين الذي نحن نتحدث عنه (ربما يكون عمر). أي تمثيل كامل من معنى الجملة يجب أن يذكر توم وعمر. نحن في أغلب الأحيان يمكن ان نكتشف هذا بالنظر إلى الجملة السابقة، ومثال على ذلك: -:

- ذهب عمر إلى المتنزه.
- هو اطعم القط الابيض.
- نحن يمكن أن نحسب من هذا بأن `` هو " يشير إلى عمر. نحن لربما أيضاً نضمن بأن القط الابيض في المتنزه، لكن لتضمنين بأنه توم نحن نحتاج إلى بعض المعلومات العامة الإضافية أو المعرفة السياقية - الـ بي تفيد بأن القط الابيض توم هو القط الابيض الوحيد الذي يتردد على المتنزه عموماً. عموماً هذا النوع من الاستدلال صعب جداً، مع ذلك تماماً يمكن أن نخذ بلستعمال الاستراتيجيات البسيطة، مثل النظر إلى "من" المذكورة في الجملة السابقة لمعرفة ان `` هو " يشير إليها. بالطبع، أحياناً هناك قد يكون شخصين (أو قطان) الذي يشير اليهما المتكلم ، ومثال على ذلك: -:
- كان هناك قط ابيض وقط أسود في المتنزه.
- ذهب عمر إلى المتنزه مع زيد.
- هو اطعم القط.

في مثل هذه الحالات نحن لثمة غموض مرجعي. من الواضح تماماً ان نادر، لكن عموماً يمكن أن يكون مشكلة كبيرة. عندما يكون المرجع المقصود غير واضح فإن نظام الحوار في اللغة الطبيعية يجب أن يبدأ بتوضيح حوار ثانوي، يسأل عن المثال `` تعني الواحد الابيض أو الأسود. " .

على أية حال، شيء آخر يقع في أغلب الأحيان في هذه المرحلة من التحليل وهو أن نحاول تخمين أهداف التعبير. على سبيل المثال، إذا سأل شخص ما، "كم الشيء ؟" نفترض عموماً بأن هدفه (من المحتمل) ان يشتريه. إذا أنت يمكن أن تحزر أهداف الناس فلنت يمكن أن تكون أكثر قدرة في الرد على أسئلتهم. لنفترض المثال التالي ، نظام خدمات معلومات الطيران الآلية، عند سؤالها عن موعد الطيران القادم إلى باريس ، فإنه لايجب فقط اعطاء الإجابة " 6 مساءً " إذا عرفت بأن هذه الرحلة كاملة. بل يجب أن يضمن بأن السائل يريد السفر ، فيصدق بأن هذا محتمل، ويقول " 6 مساءً، لكنه كامل. إن الطيران القادم بمقعد فارغ في 8 مساءً. "

توالد الجمل والمعاني

في الفطنتين السابقتين ناقشنا فهم اللغة الطبيعي فقط. على أية حال، يجب أن نكون مدركين أيضاً المشاكل في توليد اللغة الطبيعية. لنفترض اننا نريد قول شيء ما (ومثال على ذلك: -، يأكل (عمر، شوكلاته))، أو عندما لدينا هدف نريد إنجازه (ومثال على ذلك: -، الطلب من زيد إغلاق الباب)، فإن هناك عدة أشكال لإنجاز ذلك خلال اللغة:

- هو يأكل الشوكولاتة.
- هذه الشوكولاتة التي يأكلها عمر.
- يأكل عمر الشوكولاتة.
- الشوكولاته أكلت من قبل عمر.

وللمثال الثاني :

- إغلاق الباب.
- الجو بارد هنا.
- هل يمكن أن تغلق الباب.

أي نظام توليد جمل ومعاني يجب أن يكون قادر على إختيار وبشكل ملائم من بين الإنشاءات المحتملة المختلفة، والمستندة على معرفة السياق. إذا عليه كتابة نص معقد ، فيجب أن يعرف كيف يجعل ذلك النص متماسك.

التخطيط

ماذا يتضمن التخطيط ؟

تخطيط المشاكل يعتبر مشكلة كبيرة لأن :

- المشاكل ليست بديهية بالتأكيد.
- تتضمن الحلول العديد من السمات التي ناقشناها قبل، مثل :
- o البحث وإستراتيجيات حل المشكلة.

o مخططات تمثيل المعرفة.

o تحليل المشكلة -- تقسيم المشكلة إلى عناصرها البسيطة ومحاولة حلّ هذه العناصر أولاً.
رأينا بأنّه يمكن حلّ مشكلة بإعتبار الشكل الملائم لتمثيل المعرفة وإستعمال الخوارزميات لحلّ أجزاء المشكلة وأيضاً لإستعمال طرق البحث.

البحث في التخطيط

البحث كما رأينا سابقاً وبشكل اساسي هو الإنتقال من الحالة أولية إلى حالة الهدف. تقنيات البحث الكلاسيكية يُمكن أن نطبق لتخطيط الحالة في هذا الأسلوب:

خوارزمية *A

-- أفضل بحث أول،

تحليل المشكلة

-- بتكيب، مشكلة إطار.

خوارزمية *AO

-- نقرّج المشكلة إلى الأجزاء المتميّزة واضحة المعالم.

التفكيك والاستنتاج الإرشادي

-- البحث العادي بالتراجع يُمكن أن يُقدّم التفكيك لإبتكار الموجه والمساعد على الكشف كذلك مراقبة التراجع.

الطريقة الرئيسية الأولى اعتبرت الحلّ كبحث من الحالة الأولية إلى حالة الهدف عبر مدى الحالات. هناك عدّة أشكال إنتقال عبر هذا الهدى بإستعمال المشغلين و خوارزمية *A التي وصفت أفضل بحث أولي خلال الوسم البياني. هذه الطريقة جيدة للمشاكل الأسهل، لكن للمشاكل الأكثر واقعية من المُستحسن إستعمال تحليل المشكلة. هنا المشكلة تُقسّم إلى المشاكل الفرعية الأصغر ومن ثَمّا تجمع الحلول الجزئية. الخطر بإستعمال هذه الطريقة يقع عندما يُعجز الطرق الأكيدة تُصبح فاشلة ويؤدي ذلك الى رمي الحل جانباً. كيف يمكن للحلّ الجزئي أن يُحفظ وكمّ لإعادة حسابه ثانية.

مشكلة الإطار -- تقرر الاشياء التي تتغيّر والتي لا تتغير -- نطوي بعض التوجيهات التي تمكّننا من التقرير ماذا تبقى على حاله وماذا نغيّر لعلها انتقلنا من حالة الى الحالة التي بعدها. مثلاً، إذا علّوت المشكلة بتصميم الإنسان الآلي المعد لتجميع السيارات فإن المحرك المركب على الهيكل لا يؤثر على مؤخّرة السيارة في الوقت الحاضر.

خوارزمية *AO مكنّنا من مُعالجة حلّ المشاكل حيث المشكلة يُمكن أن تُقسّم إلى الأجزاء المتميّزة وبعد ذلك أعادت جميع الحلول الجزئية. على أية حال الصعوبات تُظهر إذا تفاعلت الأجزاء مع بعضها البعض. أغلب المشاكل لها بعض التفاعل وهذه هيئ ضمننا على بعض الأفكار لاعتماد المراحل او الخطوات المتتالية ؛ على سبيل المثال إذا الإنسان الآلي يجب أن يَقلّ منضدة بالأجسام التي عليها من غرفة إلى أخرى؛ أو لتحرّك صوفا من غرفة إلى أخرى والبيانو قُرب المدخل. عملية اللفظي المبذول في إعادة توحيد الحلول الجزئية في مثل هذه المشاكل تُعرف بالتخطيط. من هذه النقطة سننطلق لمناقشة دور الحاسوب في تصميم خطة كيفية حلّ المشكلة. من غير المحتمل في هذه المرحلة ان الحاسوب سيحلّ المشكلة بدون التفاعل مع الانسان.

عموماً الحاسوب يُستعمل للتقرير بالنسبة الى أفضل طريقة مُناسبة لحلّ المشكلة. في إحساس واحد هذه يُمكن أن يُترجم كمحاكاة؛ على سبيل المثال في مشكلة تنظيم هبوط الطائرات، لا يجب ان يكون هناك اكثر من طائرة على مدرج المطار في نفس الوقت. عليه عندما وجد الحاسوب افضل الحلول بناء على التحقيق الذي قام به فإنه يُمكن تطبيقه في الموقع الحقيقي.

تفترض هذه النظرية بأنّ هناك إستمرارية في طريقة الحياة. لكنها لا تستطيع وضع تقديرات للتغيير السريع. كيف هذه النظرية تهتمّ بالأحداث الغير متوقّعة مثل مكوّن معيب أو حدث مزور مثل مادتين التصقنا ببعضهم البعض. عندما نواجه عيب أو بعض الحالات مستحيلة التمييز فإنه ليس من الضروري الإستئناف لمُعظم ما حلّ بنجاح وما زال مفيّداً لنفترض طفل ما يُزيل الإبر من بلورة مُحالّة جزئياً (لم تنتهي حياكتها). تكمن المشكلة في الإستئناف من النهاية المسدودة وهذا سيحتاج الى بعض التراجع (التراجع الى بداية الخط). هذه الطريقة من الحلّ تُنبع لتخفيض مستوى التعقيد ولذا لضمان معالجة ناجحة يجب أن نُقدّم التفكّك للمُساعدة في التراجع المطلوب للإهتمام بالعيوب. للمُساعدة في السيطرة على التراجع هناك العديد من الطرق تتعامل بشكل خلفي من الهدف إلى الحالة الأولية.

أمثلة تخطيط الكتل

ما هو عالم الكتل؟ -- العالم يَشتمَلُ على:

- السطح المستوي مثل سطح المنضدة
 - مجموعة كافية من الكتل المماثلة التي مُميّزت بحروف.
 - الكتل يُمكنُ أَنْ تُكَدَسَ واحد فوق الآخر لتشكيل أبراج ذات إرتفاع غير محدود.
 - التكديس انجز بـ استعمال ذراع آلي يملك عمليات أساسية وحالات يُمكنُ أَنْ يَفْعَمَ بـ استعمال المنطق و تدمج بـ استعمال العمليات المنطقية.
 - الإنسان الآلي يُمكنُ أَنْ يَحْمَلَ كتلة واحدة كل مرة فقط كتلة واحدة يُمكنُ أَنْ تُحَرَّكَ في نفس الوقت.
- نحن سَنَسْتَعْمَلُ الأعمال الأربعة:

UNSTACK(A,B)

-- يَلْتَقِطُ كتلة واضحة A من الكتلة B؛

STACK(A,B)

-- ضع الكتلة A باستعمال الذراع فوق الكتلة الواضحة B؛

PICKUP(A)

-- ارفع الكتلة A الواضحة بالذراع الفارغ؛

PUTDOWN(A)

-- ضِعْ الكتلة المَحْمُولَة A في المكان الفارغ على المنضدة.

والمسندات الخمسة:

ON(A,B)

-- الكتلة A على الكتلة B .

ONTABLE(A)

-- الكتلة A على المنضدة.

CLEAR(A)

-- الكتلة A لَيْسَ عليها شيء.

HOLDING(A)

-- الذراع يَحْمَلُ الكتلة A .

ARMEMPTY

-- الذراع لا يَحْمَلُ شيء.

بـ استعمال المنطق ولكن لَيْسَ الترميز المنطقي يُمكنُ أَنْ نَقُولَ بأنّ إذا الذراع يَحْمَلُ كتلة فهو لَيْسَ فارغاً و إذا الكتلة A على المنضدة يعني لَيْسَ أي كتلة أخرى وإذا الكتلة A على الكتلة B، يعني الكتلة B لَيْسَتْ واضحة. لماذا نَسْتَعْمَلُ عالم الكتل كَمثال؟
إنّ عالم الكتل مُختارُ لأن:

- بسيط بما فيه الكفاية ويَصَرِّفَ بشكل حسن.
 - مفهوم بسهولة
 - رغم ذلك ما زال يُزوّد ببيئة عيّلات جيدة لدراسة التخطيط:
- o المشاكل يُمكنُ أَنْ تُؤَسَّرَ الى مشاكل فرعية مُتميّزة تقريباً
- o نحن يُمكنُ أَنْ نَرى كيف ان الحلول الجزئية من الضّروري أَنْ تُدمَجَ لتشكيل حلّ كامل واقعي.

تخطيط مكونات النظام

مهام حل المشكلة البسيطة تتضمّن الوظائف التالية:

1. إختَر أفضل قاعدة (قانون) مستندة على التوجيه المساعد.
2. طبّق هذه القاعدة لَخْلُق حالة جديدة.
3. إكتشف عندما يوجد حلّ.
4. إكتشف النهايات المسدودة لتفاديهم.

لحل المشكلة الأكثر تعقيداً في أغلب الأحيان تضاف مهمة خامسة:
5. إكتشف عندما تكون الحالة قريبة الحل و إستعمل الطرق الخاصة لجعلها الحالة المحلولة.
الآن دعنا نرى ماذا تستعمل تقنيات الذكاء الاصطناعي عموماً في كل من المهام أعلاه. نحن بعد ذلك سننظر الطرق المعينة للتطبيق.

إختيار أفضل قاعدة

الطرق المستعملة تتضمن:

- إيجاد الاختلافات بين الوضعية الحالية وحالات الهدف.
- إختيار القواعد التي تقلل هذه الاختلافات بفاعلية أكثر.
- تحليل نهائيت مینز مثال جيد د.
- إذا نحن نريد السفر بالسيارة لزيارة صديق ما
- أول شيء يجب أن نحلل السيارة بالوقوف.
- إذا ليس لدينا سيارة فأنا نحتاج لواحدة.
- الاختلاف الأكبر يجب أن يُعالج أولاً.

تطبيق قاعدة

- القواعد السابقة يُمكن أن تُطبق بدون أي صعوبة كأنظمة كاملة حُدثت وقواعد مكنّت النظام للتقدم من حالة لأخرى.
- الآن يجب أن نكون قادرين على مُعالجة القواعد التي تغطي فقط أجزاء من الأنظمة.
- استعمل عدد من النظريات لهذه المهمة.

نظرية Green's (1969)

أساساً هذا يُصرّح بأننا نلاحظ التغييرات على الحالة والنتيجة من تطبيق القاعدة.

لنفترض مشكلة كتلتين A و B والمكدسة فوق بعضهم البعض (A في القمة). لربما نحن عندنا حالة أولية S_0 التي يُمكن أن تُوصف كالتالي:

$$ONTABLE(A, B, S_0) \wedge CLEAR(A, S_0)$$

إذا اردنا ان $UNSTACK(A, B)$ ، وضع A على B. نبدو العملية كالتالي:

$$CLEAR(x, s) \wedge ON(x, y, s) \rightarrow [HOLDING(x, DO(UNSTACK(x, y), s)) \wedge CLEAR(y, DO(UNSTACK(x, y), s))]$$

حيث x, y أي كتل ، s أي حالة و $DO()$ نخذ النتائج الوسمية الجديدة من العمل المُعطى.

نتيجة وضع هذه على حالة S_0 لإعطاء الحالة S_1 فإننا نحصل على:

$$HOLDING(A, S_1) \wedge CLEAR(B, S_1).$$

هناك بضعة مشاكل بهذه النظرية:

مشكلة الإطار

-- من الوارد أعلاه نحن نعرف بأن B ما زالت على المنضدة. هذا من الضروري أن يُفكر إلى بديهيات الإطار التي تُصِفُ مكونات الحالة والتي ليست متأثرة بالعمل.

مشكلة المؤهل

-- إذا حللنا مشكلة الإطار ، فلن الوصف الناتج قد يكون ناقص. هل نحتاج لتشفير ان الكتلة لا يمكن أن تكون موضوعاً على نفسها؟ إذا كان الأمر كذلك هل يجب أن تفشل هذه المحاولة؟ إذا سُمحنا بالفشل فإن الأمور ستتعدد -- هل سَنُسمح بالكثير من الأحداث الغير محتملة؟

مشكلة النتيجة

-- بعد وضع الكتلة A، مسبقاً، كيف نعرف بأن A لم نعد في موقعه الأولي؟
ليس فقط الصعوبة بتحديد بالضبط الأحداث التي لم تقع (مشكلة إطار) بل بتحديد الذي يحدث.

نظرية STRIPS (1971)

STRIPS اقترح نظرياً أخرى تقوم على:

- مبدائنا كل مشغل لهي ثلاثة قوائم من المسندات المرتبطة معه:
o قائمة الأشياء التي تُصبح "صح" وتسمى Add.
o قائمة الأشياء التي تُصبح "خاطي" وتسمى Delete.
o مجموعة الشروط التي يجب أن تكون حقيقة قبل تطبيق المشغل.
- أي شيء ليس في هذه القوائم، مفترض أن يكون غير متأثرة بالعملية.
- هذه الطريقة هي التطبيق الأولي لـ STRIPS - وسع ليعتبر الأشكال الأخرى من التفكيك / التخطيط (ومثال على ذلك: - الطرق الرتبية، تخطيط مجموعة الاهداف و حتى التخطيط الاخطي -- يرى لاحقاً)

لنأخذ المثال التالي في عمليات الكتل العالمية والاساسية:

STACK

-- يتطلب الذراع لكي يحمل الكتلة A ، والكتلة الأخرى B لكي تكون واضحة. بعد ذلك الكتلة A على الكتلة B والذراع تصبح فارغة وهذه حقيقة -- ADD ؛ الذراع لا يحمل كتلة و الكتلة B ليست واضحة؛ المسندات الفاشلة تحذف Delete ؛

UNSTACK

-- يتطلب بأن الكتلة A على الكتلة B؛ والذراع فارغ و الكتلة A واضحة. بعد ذلك الكتلة B واضحة والذراع يحمل كتلة A - ADD ؛ الذراع ليس فارغ والكتلة A ليست على الكتلة B -- DELETE ؛

وهكذا نحن قللنا المعلومات التي من الضروري أن تحمل. إذا الخاصية الجديدة يفيد بأننا لسنا بحاجة إلى أن تُضيف بديهيات جديدة لإيجاد المشغلين. على خلاف ذلك في طريقة Greens ، نزيل مؤشر الحالة ونستعمل قاعدة بيانات المسندات للإشارة إلى الوضعية الحالية وعليه إذا الحالة الأخيرة كانت:

ONTABLE(B) ^ON(A,B) ^CLEAR(A)

بعد عملية UNSTACK ، الحالة الجديدة ستكون :

ONTABLE(B) ^CLEAR(B) ^HOLDING(A) ^CLEAR(A)

إكتشاف التقدّم

الحل النهائي يُمكن أن يُكتشف إذا :

- نحن يُمكن أن نبتكر المسند الذي يكون حقيقي (صح) عند وجود الحل وخاطي بخلاف ذلك.
- يتطلب الكثير من التفكيك ويتطلب البرهان.
- إكتشاف الآثار الخاطئة ضرورية أيضاً:
- ومثال على ذلك: - بحث A* -- إذا كان التقدّم غير كافي فإن هذا الأثر يلغي لمصلحة واحد أكثر تفانلاً.
- أحياناً من الواضح أن حل المشكلة بطريق واحد خفض المشكلة إلى الأجزاء التي تعتبر أصعب من الحالة الأصلية.
- بالرّجوع من حالة الهدف إلى الحالة الأولية من المحتمل إكتشاف تناقضات وأي أثر أو طريق الذي تتضمن نزاع يُمكن أن يتسبب بالخروج من البحث.
- تخفيهن عدد الطرق المحتملة يعني بأن هناك مصادر أكثر متوفرة لأولئك الذين اهتموا.

لنفترض بأن معلّم الحاسوب مريض. في المدرسة هناك بديلان محتملان:

- نقل معلّم الرياضيات والذي يعرف استعمال الحاسبات أو
- تجلب معلم آخر.

المشاكل المحتملة:

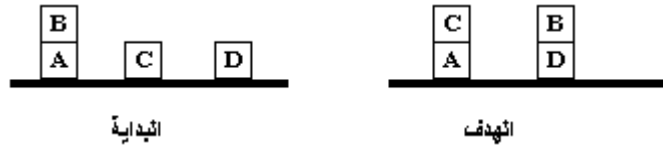
- إذا مدرس الرياضيات هو المعلّم الوحيد للرياضيات فإن المشكلة لم تُحل.
 - إذا بتبين انه ليس هناك مال لدى المدرسة فإن الحل الثاني يُمكن أن يكون مستحيل.
- إذا المشاكل قابلة للتقسيم الى عناصرها الاساسية فنحن يُمكن أن نُعالجهم على اساس عناصر وبعد ذلك يُهم ربطهم، هكذا؟

لنعتبر اننا وصلنا الى الحالة النهائية بمعالجة المشكلة ك عناصر في الوضعية الحالية ونلاحظ الاختلافات بين حالة الهدف والوضعية الحالية وحالة الهدف والحالة الأولى ونستخدم تقنيات تحليل نهائية مبرز للتحرك في أفضل اتجاه. الأفضل هو العمل خلفا في الطريق الذي يؤدي إلى الوضعية الحالية و روى إذا كان هناك خيارات. هو قد يكون الطريق الاختياري الوح ي الذي يُمكن أن يؤدي إلى حل بينما أدى الطريق الحالي إلى تناقض. عموماً هذا يعني بأن بعض الشروط تغيّرت قبل أخذ الطريق الاختياري خلال المشكلة. تتضمن النظرية الأخرى تأجيل قرارات حتى واحد يجب أن، يترك إتخاذ القرارات حتى المزيد من المعلومات طرق متوفرة وأخرى استكشفت. في أغلب الأحيان بعض القرارات ليس من الضروري أن تأخذ كهذه العقدة ما وصلت.

تخطيط مجموعة الهدف

الفكرة الأساسية هي معالجة إستعمالات الأهداف المركبة التفاعلية باستخدام مجموعة الاهداف، هنا المجموعة تحتوي على :

- الأهداف،
 - الأوامر -- إضافته حذف وشروط القوائم
 - قاعدة البيانات تبقى الحالة الحالية لكل امر إستعمل.
- لنفترض التالي : حيث نريد المضي من حالة البداية الى حالة الهدف.



الوسم . 22 مثال تخطيط مجموعة الهدف

نحن يُمكن أن نصِف حالة البداية:

$ON(B, A) \wedge ONTABLE(A) \wedge ONTABLE(C) \wedge ONTABLE(D) \wedge ARMEMPTY$

وحالة الهدف:

$ON(C, A) \wedge ON(B, D) \wedge ONTABLE(A) \wedge ONTABLE(D)$

- مبدائيا مجموعة الهدف هي حالة الهدف.
- نحن قسمنا المشكلة إلى أربعة مشاكل فرعية
- إثنتين منها محلولة لانهم حقيقيون في الحالة الأولى. -- $ONTABLE(A), ONTABLE(D)$.
- بالنسبة للإثنتين الاخرين -- هناك طريقان للمضي:

- $ON(C, A)$
-


```

9.      ^ ARMEMPTY
10.
11.  UNSTACK (C, x)
12.
13.      CLEAR (A) ^ HOLDING (C)
14.
15.      STACK (C, A)
16.
17.      ON (B, D)
18.
19.      ON (C, A) ^ ON (B, D) ^ ONTABLE (A)
20.
21.      ^
22.  ONTABLE (D)
23.
24.

```

```

1
1.  ONTABLE (C)
2.
3.  CLEAR (C)
4.
5.  ARMEMPTY
6.
7.  ONTABLE (C) ^ CLEAR (C)
8.
9.      ^ ARMEMPTY
10.
11.  PICKUP (C)
12.
13.      CLEAR (A) ^ HOLDING (C)
14.
15.      STACK (C, A)
16.
17.      ON (B, D)
18.
19.      ON (C, A) ^ ON (B, D) ^ ONTABLE (A)
20.
21.      ^
22.  ONTABLE (D)
23.
24.

```

في الطريق الأول نحن يُمكن أن نرى ثلاث إشارات إلى بعض الكتلة ، x وهذا يجب أن يُشير إلى نفس الكتلة، بالرغم من أنه خلال البحث من المعقول ان عِدّة كُتَل سَتُصبح ملحقة بشكل مؤقت. لذلك ربط المتغيرات بالكُتَل يجب أن يسجل. التّحقيق التطبيق الهدف الأول يتطلّب تكديس C على بعض الكتلة الواضحة.

CLEAR (x)

HOLDING (C)

CLEAR (x) ^ HOLDING (C)

STACK (C, x)

CLEAR (C)

ARMEMPTY

نُلاحظُ الآن بأنَّ أحدَ الأهدافِ المُخلَّقةِ HOLDING(C) الذي كَانَ الهدفَ الذي نحنُ كُنَّا نُحاولُ إنجازَ ه بتطبيقِ UNSTACK(C, some block) في هذه الحالةِ PICKUP(C) في النظريةِ الأخرى. لذا هو يَظهرُ بأنَّنا أضفنا أهدافَ جديدةً ولم نحقق تقدّمَ ومن ناحيةِ خوارزميةِ A* يبدو أفضلَ لمحاولةِ النظريةِ الأخرى. بالرجوعِ إلى النظريةِ الثانيةِ :

- نحنُ يُمكنُ أن نرى بأنَّ الهدفَ الأولَ ينجز ,وضع الكتلةِ C على المنضدةِ.
 - الهدفَ الثاني ينجزُ، حيثُ الكتلةِ C واضحة.
 - تذكرُ بأنَّ HOLDING(B) ما زالتْ صريحةً وهذا يعني بأنَّ الذراعَ ليسَ فارغاً. هذا يُمكنُ أن ينجزَ بوضعِ B على المنضدةِ أو يكدسه على الكتلةِ D بشرطِ ان واضح.
 - للنظرِ الآن هل يُمكنُ هنا إستعمالُ مُقارَنةِ القوائمِ Add للمشغلينِ المتنافسينِ مع الأهدافِ في مجموعةِ الهدفِ وهناك مطابقةٌ معَ ON(B,D) التي تطبقُ بـ STACK (B,D). هذا أيضاً يُبطِئُ بعضَ الكتلِ الى الكتلةِ D.
 - إستعمالُ STACK (B,D) يُولّدُ أهدافَ إضافيةً CLEAR(D) وHOLDING(B) .
- اذن مجموعةِ الهدفِ الجديدةِ تُصبحُ؛

CLEAR (D)

HOLDING (B)

CLEAR (D) \wedge HOLDING (B)

STACK (B, D)

ONTABLE (C) \wedge CLEAR (C) \wedge ARMEMPTY

PICKUP (C)

في هذه النقطةِ، الهدفَ الأعلى حقيقيً والقبالي ايضاً وهكذا الهدفَ المشتركَ يُؤدّي إلى تطبيقِ STACK (B,D)، الذي يعني بأنَّ النموذجَ العامَ يُصبحُ

Ontable ONTABLE(A) \wedge ONTABLE(C) \wedge ONTABLE(D) \wedge ON(B,D) \wedge ARMEMPTY

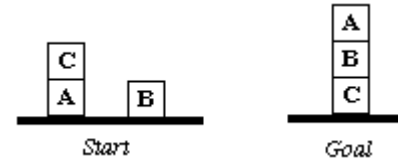
هذا يعني بأنَّنا يُمكنُ أن نُؤدّي PICKUP(C) وبعد ذلكِ STACK (C,A).
نأتي الآن إلى الهدفِ ON(B,D) ندركُ بأنَّ هذا كَانَ مُنجَزَ ولتدقيقِ الهدفِ النهائي نَسْتَقُ الخطةَ التاليةَ:

1. UNSTACK(B,A)
2. STACK (B,D)
3. PICKUP(C)
4. STACK (C,A)

هذه الطريقةُ تُنتجُ الخطةَ التي تَسْتَعْمَلُ تقنياتَ الفكاءِ الاصطناعي الحيدة مثل الموجه المساعد لإيجاد الأهدافِ المماثلةِ و خوارزميةِ A* لإكتشافِ الطرقِ الغير مرجوةِ النجاحِ التي يُمكنُ أن تُنبَذَ.

Sussman Anomaly (1975)

الطريقة السابقة قد تُخفق في إعطاء حل جيد. لنعتبر:



الوسم. 25 Sussman's Anomaly
إنّ حالة البداية مُعطية من قبل:

$ON(C, A) \wedge ONTABLE(A) \wedge ONTABLE(B) \wedge ARMEMPTY$

والهدف من قبل:

$ON(A, B) \wedge ON(B, C)$

هذا يُؤدّي إلى طريقتين حسب التالي:

1. $ON(A, B)$
- 2.
3. $ON(B, C)$
- 4.
5. $ON(A, B) \wedge ON(B, C)$
- 6.
- 7.
8. $ON(B, C)$
- 9.
10. $ON(A, B)$
- 11.
12. $ON(A, B) \wedge ON(B, C)$
- 13.
- 14.

إختيار الطريق 1 ومحاولة الحصول على "الكتلة A على الكتلة B" يُؤدّي إلى مجموعة الهدف:

$ON(C, A)$

$CLEAR(C)$

$ARMEMPTY$

$ON(C, A) \wedge CLEAR(C) \wedge ARMEMPTY$

$UNSTACK(C, A)$

$ARMEMPTY$

$CLEAR(A) \wedge ARMEMPTY$

$PICKUP(A)$

CLEAR (B) \wedge HOLDING (A)

STACK (A, B)

ON (B, C)

ON (A, B) \wedge ON (B, C)

هذا يُنجز كتلة A على الكتلة B والتي أُنتجت بوضع الكتلة C على المنضدة.
إنّ سلسلة المشغلين :

1. UNSTACK(C,A)
2. PUTDOWN(C)
3. PICKUP(A)
4. STACK (A,B)

العمل على الهدف القادم ON(B,C) يتطلّب الكتلة B لكي تُوضّح لكي يُمكن تكدّسها على الكتلة C . لسوء الحظ نحتاج إلى رفع الكتلة A علماً بأننا قد رفعناها للتو. و هكذا قائمة المشغلين تُصبح

1. UNSTACK(C,A)
2. PUTDOWN(C)
3. PICKUP(A)
4. STACK (A,B)
5. UNSTACK(A,B)
6. PUTDOWN(A)
7. PICKUP(B)
8. STACK (B,C)

للوصول إلى حالة ان الكتلة A ليست على الكتلة B فإنه يتطلب عمليتان إضافيتان:

1. PICKUP(A)
2. STACK(A,B)

بتحليل هذه السلسلة نلاحظ ان :
الخطوات 4 و 5 نظراء ويلغي أحدهما الآخر ،
الخطوات 3 ، 6 نظراء ويلغي أحدهما الآخر.
لذا المخطط الأكثر كفاءة:

1. UNSTACK(C,A)
2. PUTDOWN(C)
3. PICKUP(B)
4. STACK (B,C)
5. PICKUP(A)
6. STACK(A,B)

للإنتاج في مثل كل هذه الحالات فإن هذا المخطط الكفوء حيث هذا التفاعل بين الأهداف يتطلب تقنيات أكثر تطوراً التي سنتقش في الفصل القادم.

التخطيط اللاحطي باستعمال الإيداع المقيد

دعنا نُعيد النظر في نظرية SUSSMAN ANOMALY

- مشاكل مثل هذه تتطلب مشاكل فرعية للعمل عليها بشكل آني.
 - خطة الاحطى تستعمل موجهاً مساعدة مثل:
 - 1. حاول إنجاز $ON(A,B)$ بتتطيف كتلة A ووضع كتلة C على المنضدة.
 - 2. أنجز $ON(B,C)$ بتكديس الكتلة B على الكتلة C .
 - 3. أتم $ON(A,B)$ بتكديس الكتلة A على الكتلة B .
 - الإيداع المقيد يبدو كتقنية مركزية في أنظمة التخطيط الأخيرة .
 - الإيداع المقيد يُعزّز الخطة ب:
 - اقتراح مشغلين،
 - محاولة طلبهم، و
 - اعطاء ملزمات بين المتغيرات في المشغلين والكتل الفعلية.
- الخطة الأولية لن تشمل أي خطوات وبدراسة حالة الهدف يمكن للخطوات المحتملة ان تُؤد.
- ليس هناك طلب أو تفصيل في هذه المرحلة.
- بشكل تدريجي تفاصيل أكثر ستعطى وقيود حول طلب المجموعات الفرعية للخطوات سقدّم حتى الوصول الى تخليق سلسلة منظمة كاملة.
- في هذه المشكلة، تحليل طرفيات الوسائل، تفرخ خطوتين بشروط النهاية $ON(A,B)$ و $ON(B,C)$ والتي تُشير ان إلى مجموعة المشغل (STACK) وتُعطي النتائج الموضحة اسفلا، حيث أن المشغل سيقب بشروطها المسبقة واتبعت بشروط الايداع:

CLEAR (B)

CLEAR (C)

*HOLDING (A)

*HOLDING (B)

STACK (A, B)

STACK (B, C)

ARMEMPTY

ARMEMPTY

ON (A, B)

ON (B, C)

¬ CLEAR (B)

¬ CLEAR (C)

¬ HOLDING (A)

¬ HOLDING (B)

ملاحظات:

- ليس هناك طلب (امر تشغيل) في هذه المرحلة.
- شروط مسبقة غير مُنجزّة مع علامة (*).
- لى من شروط HOLDING المسبقة غير مُنجز حيث أن الذراع لا يحمل شيء في الحالة الأولية.
- حذف شروط الايداع محدد بعلامة (¬).

قدّم العديد من طرق التخطيط الموجهات المساعدة لتحقيق الأهداف أو الشروط المسبقة. وضعت طريقة تخطيط TWEAK كل هذا سوية تحت شكلية واحدة. الطرق الأخرى التي قدّمت / إستعملت الموجهات المساعدة المذكورة في القسم التالي.

موجهات Tweak ليستعمال الإيداع المقيد

تتضمن طريقة تخطيط القرص الموجه التالي.

Step Addition إضافة خطوة

-- يخلق خطوات جديدة (GPS).

Promotion الترقية

-- يرغم الخطوة للذهاب قبل الخطوة الأخرى (Sussman HACKER).

Declobbering

-- يضع خطوة جديدة بين خطوتين لإعادة شرط مسبق (NOAH, NONLIN).

Simple Establishment المؤسسة البسيطة

-- يُخصّص قيمة إلى متغير لضمان شرط مسبق (TWEAK).

Separation الإفتراق

-- منع متغيرات أن تخصّص بعض القيم (TWEAK).

لننظر الآن إلى تأثير كل موجه إرشادي. كما يجب أن نحاول وننجز الشروط المسبقة من عملية المجموعة اعلاه. يمكن أن نحاول رفع الكتل الخاصة:

CLEAR (A)

CLEAR (C)

ONTABLE (A)

ONTABLE (B)

*ARMEMPTY

*ARMEMPTY

PICKUP (A)

PICKUP (B)

¬ONTABLE (A)

¬ONTABLE (B)

¬ARMEMPTY

¬ARMEMPTY

HOLDING (A)

HOLDING (B)

في الوقت الحاضر ليس هناك خطة كالإيداع المقيد لهذه المجموعة يمكن أن تنفي شروط مسبقة لخطة المجموعة (STACK) الأولى، لذا نحن يجب أن نعطي امر تشغيل حسب التالي:

- إذا الخطة النهائية تحتوي على PICKUP ثم خطوة STACK فإن
- HOLDING - شروط مسبقة من الضروري أن نطبق باستخدام خطوات أخرى.
- حل هذا بفرض الطلب وذلك بتقديم القيود حينما يثبت أن الإضافة مُستخدمة.
- في هذه الحالة نحتاج للذكر أن خطوة PICKUP يجب أن تسبق تطبيق خطوة STACK. بمعنى آخر

PICKUP(A) ← STACK(A,B)

PICKUP(B) ← STACK(B,C)

هذا يعطي أربع خطوات منظّمة جزئياً وأربعة شروط غير مُنجزّة

- * CLEAR(A) -- كتلة A ليست واضحة في الحالة الأولية.
 - * CLEAR(B) -- بالرغم من أن الكتلة B واضحة في المجموعة الرسمية الأولية STACK(A,B) مع شروط الإيداع \neg CLEAR(B) قد تسبق الخطوة * CLEAR(B) مع شرط مسبق.
 - إثبات * ARMEMPTY -- حالة أولي تجعل ARMEMPTY لكن خطوة PICKUP لها \neg ARMEMPTY ويمكن أن تسبق هذه الخطوة ثانية.
- نحن يمكن أن نستعمل الترقية الإرشادي لإجبار مشغل واحد لسبق آخر، حيث شروط الإيداع لمشغل واحد
 STACK(A,B) لا تنفي الشرط المسبق CLEAR(B) من المشغل الآخر PICKUP(B). هذا الطلب ممثل بـ:
 $PICKUP(B) \leftarrow STACK(A,B)$
- نحن يمكن أن نستعمل الترقية لإنجاز إحدى شروط ARMEMPTY المسبقة:
 جعل PICKUP(B) يسبق PICKUP(A) تضمن بأن الذراع فارغ وكل الشروط لـ PICKUP(B) تجتمع.
 هذا يكتب كالتالي :

$PICKUP(B) \leftarrow PICKUP(A).$

لسوء الحظ شروط الإيداع للمشغل الأول هو ان الذراع يصبح غير فارغ ، لذا نحتاج لإستعمال الموجه الإرشادي المرتبط (المرقع) لإنجاز الشروط المسبقة للمشغل الثاني PICKUP(A).
 الترقية

- PICKUP(B) تؤكد \neg ARMEMPTY.
- لكن إذا ادخلنا خطوة بين PICKUP(A) و PICKUP(B) لإعادة تأكيد ARMEMPTY فإنه يمكن أن ننجز الشرط المسبق.
- STACK(B,C) يمكن أن تعمل هذه لذا نودع قيداً آخرًا:

$PICKUP(B) \leftarrow STACK(B,C) \leftarrow PICKUP(A)$

نحن ما زلنا نحتاج لإنجاز CLEAR(A) :
 إن المشغل الملائم هو UNSTACK(x,A) بإضافة الخطوة. هذا يؤدي إلى المجموعة التالية للشروط
 *CLEAR (x)

*ON (x, A)

*ARMEMPTY

UNSTACK (x, A)

\neg ON (x, A)

\neg ARMEMPTY

HOLDING (x)

CLEAR (A)

المتغير x يمكن أن يكون مربوط بالكتلة C بالمؤسسة البسيطة الإرشادية حيث C على A في الحالة الأولية.
 إن الشروط المسبقة CLEAR(C) و ARMEMPTY متفقي بـ STACK(B,C) و بـ PICKUP(B) أو PICKUP(A) على أية حال.

لذا نحن يَجِبُ أَنْ نُقَدِّمَ ثلاثة أوامر بالترقية لضمان المشغل UNSTACK(C,A).
 $UNSTACK(C,A) \leftarrow STACK(B,C)$

$UNSTACK(C,A) \leftarrow PICKUP(A)$

$UNSTACK(C,A) \leftarrow PICKUP(B)$

تتضمنُ الترقيةُ إضافة خطوة وهذا يضربُ أحد الشروط المسبقة PICKUP(B) بمعنى ARMEMPTY ، دائماً
 المشكلة المحتملة مع هذه الإرشادية.

على أية حال لم نفقد الكُل ، حيث مازال هناك مشغل ، PUTDOWN الذي عنده شروط الإيداع المطلوب ومعطى
 ان المشغل UNSTACK(C,A) ولَدَ الشرط المسبق له مِنْ HOLDING(C) لذا نحن يُمكنُ أَنْ نُنتِجَ مشغل
 إضافي بنجاح

HOLDING (C)

PUTDOWN (C)

\neg HOLDING (C)

ONTABLE (C)

ARMEMPTY

هذا المشغل يرفع المشغل PICKUP(B) و يُنتِجُ السلسلة :

$UNSTACK(C,A) \leftarrow PICKUP(B) \leftarrow PUTDOWN(C)$

هذا يُنتِجُ السلسلة النهائية:

1. UNSTACK(C,A)
2. PUTDOWN(C)
3. PICKUP(B)
4. STACK (B,C)
5. PICKUP(A)
6. STACK(A,B)

دعنا نُنهي هذا القسم بالنظر إلى خوارزمية TWEAK:

1. عرف S ليُكون مجموعة المقترحات في حالة الهدف.
2. الإعادة

1. احذف بعض المقترحات الغير مُنجزَة P مِنْ S .
2. أنجز P باستعمال أحد الموجهات الارشادية.
3. راجع كُل الخطوات، من ضمن ذلك الخطوات المضافة، لإيجاد كُل الشروط المسبقة الغير المُنجزَة،
 أضف هذا إلى S مجموعة الشروط المسبقة الغير المُنجزَة.
 حتى تصبح المجموعة S فارغة (شرط الاعادة).
3. أكمل الخطة بتحويل الطلبات الجزئية إلى طلب كلي يُؤدّي كُل المتطلبات الضرورية.

التَّعَلُّمُ

ما هو التَّعَلُّمُ؟

التَّعَلُّمُ يعتبر جزء مهم من الذكاء الاصطناعي ، ربما لدرجة أكبر مِنْ التخطيط حيث ان:

- المشاكل صعبة -- أصعب مِنْ المشاكل في التخطيط.
 - الحلول المتعرف عليها ليست مشتركة لهما في التخطيط.
 - هدف الذكاء الاصطناعي هو نقلتي الحاسبات من التَّعَلُّمُ بدلاً مِنْ التَّوَمُّجَةِ.
- التَّعَلُّمُ هو الجزء من الذكاء الاصطناعي الذي يركِّزُ على عمليات التحسين الذاتية. م عالِجُ المعلومات التي تُحسَّنُ أدائهم أو تُكَبِّرُ قواعد معرفتهم تَعْنِي التَّعَلُّمُ.
- لماذا التعلم صعب؟

- يُشِيرُ الذكاء ضمناً إلى أَنَّ الكائن الحي أو الآلة يَجِبُ أَنْ يَكُونَ قادرين على التَّكَيُّفِ مع الحالات الجديدة.
- الانسان او الآلة يَجِبُ أَنْ يَكُونَ قادر على التَّعَلُّمُ لِيَعْمَلَ أشياء جديدة.
- هذا يَتَطَلَّبُ إكتساب المعرفة، الإستدلال والاستنتاج، تَعْدِيث / بثقة قاعدة المعرفة، إكتساب الموجه المساعد ، تطبيقي عمليات بحث أسرع، الخ.

كَيْفَ نَتَعَلَّمُ؟

العديد مِنْ النظريَّاتِ طرحتْ للمُحاوَلَةِ بتَّوِيدِ الآلة بقبليَّة التعلم. هذا لأن مهام التَّعَلُّمُ تغطي تشكيلة واسعة من الظواهر.

نخرج هنا عدة أمثلة عن كيفية التَّعَلُّمُ. نحن سنناقش هذا بالتفصيل لاحقاً:

مِهَارَةُ التَّنْقِيَةِ وَالتَّصْفِيَةِ

-- التَّعَلُّمُ بِالْمُمارَسَةِ، ومثال على ذلك: - العزف على البيانو.

إِكْتِسَابُ الْمَعْرِفَةِ

-- التَّعَلُّمُ بالتَّجَرِبَةِ وبتَّعَرُّي الخبرات الناتجة عن التَّجَرِبَةِ في قاعدة المعرفة. مثال على ذلك حفظ البصم.

الْأَخْذُ بِالنَّصِيحَةِ

-- مشابه للتَّعَلُّمُ بالتَّكرار بالرغم من أن المعرفة المُدخلة قَدْ تحتاج الى أَنْ تُحوَّلَ (يتم اجراء بعض عمليات التحويل) لإستخدامها بشكل عملي فعال.

حَلُّ الْمَشْكِلةِ

-- إذا حُلِّتْ مشكلة ما فإن احد ما قَدْ يَتَعَلَّمُ مِنْ هذه التَّجَرِبَةِ (التعلم من اخطاء الغير). في المرة القادمة عندما نواجه مشكلة مماثلة فإنه يُمكن أَنْ نَحْلُها بكفاءة أكبر. هذا لا يَتَضَمَّنُ معرفة جديدة متراكمة عادة لكن قَدْ يَتَضَمَّنُ إعادة تنظيم البيانات أو التذكُّر كَيْفَ نصل إلى الحَلِّ.

الإِسْتِقْرَاءُ

-- يُمكنُ التَّعَلُّمُ مِنْ الأمثلة. يُصنَّفُ البشرُ الأشياء في الحياة في أغلب الأحيان بدون معرفة قواعد واضحة. يَتَطَلَّبُ هذا النوع من التعلم عادة وجود معلم أو مدرِّب للمُساعدَةِ على التصنيف.

الإِكْتِشَافُ

-- تَعَلُّمُ المعرفة بدون مساعدة معلم.

التَّنَاضُرُ

-- إذا كان النظام يُمكنُ أَنْ يَهِيزَ التشابه والتماثل في المعلومات المخزَّنة لديه فإنه قَدْ يَكُونُ قادر على تحويل بعض المعرفة لِتَحْسِينِ حَلِّ المهمة التي في متناول اليد.

التَّعَلُّمُ بِالرُّوتِينِ (التَّكَرُّارِ)

التَّعَلُّمُ بالروتينين أساساً هو إستظهارُ (حفظ بصم)، بالخصائص التالية:

- حفظ المعرفة لِإِسْتَعْمَالِها ثانيةً.
- إسترجاع المعلومات هو المشكلة الوحيدة.
- لا ضرورة للحساب المتكرَّر أو الإستدلال أو الإستفسارات.

مثال بسيط التعلّم بالتكرار هو الذاكرة الوسيطة بالكمبيوتر

- تخزين القيم المحسوبة (أو قطع كبيرة من البيانات)
 - إعادة استدعاء هذه المعلومات عندما يُطلب الحساب ذلك.
 - يمكن توفير وقت هام.
 - الكثير من برامج الذكاء الاصطناعي، إستخدمت الذاكرة الوسيطة بشكل فعال.
- الإستظهار هو ضرورة رئيسية للتعلّم:

- هو ضرورة أساسية لأي برنامج ذكي - هل هو عملية تعلم منفصلة؟
 - الإستظهار يُمكن أن يكون موضوع معقد -- ما هي أفضل طريقة لتخزين المعرفة؟
 - إستخدم برنامج التدقيق "مدقق صموئيل" التعلّم بالتكرار (إستعمل أيضا عامل التعديل الذي سَنُناقش لاحقاً).
 - بحث "الأكبر- الأصغر" (minimax) إستعمل لإستكشاف شجرة اللعبة.
 - قيود الوقت لا تسمح بعمليات بحث كاملة.
 - يُسجل مواقع اللوحة ويحسب النقاط المُحرزة في نهايات البحث.
 - إذا ظهر نفس موقع اللوحة لاحقاً في اللعبة، فإن القيمة المُخزنة يُمكن أن تستدعى والتأثير النهائي هو وقوع البحث الأكثر عمقاً.
- التعلّم بالإستظهار أساساً هو عملية بسيطة. على أية حال هو يُصور بعض القضايا ذات ال صلة بقضايا التعلّم الأكثر تعقيداً.

المنظم

-- الوصول الى القيمة المُخزنة يجب أن يكون أسرع من إعادة حسائها. طرق مثل الفهرسة والتصنيف يُمكن أن يُستخدما للتأمين من تنفيذ هذا.

التعميم

-- عدد الأجسام المُخزنة فعلاً يُمكن أن يكون كبير جداً. نحن قد نحتاج لتعميم بعض المعلومات لجعل المشكلة سهلة الانقياد.

إستقرار البيئة

-- التعلّم بالتكرار ليس فعال جداً في بيئة سريعة التغيير. إذا البيئة تتغير فإنه يجب أن نكتشف ونُسجل بالضبط ما الذي تغير -- مشكلة الإطار.

تخزين قيم الحساب

تعلّم الإستظهار لا يجب أن ينقص كفاءة النظام. يجب ان نكون قادرين على التفرير فيما اذا هو اسوأ تخزني القيمة في المركز الأول. لنعتبر حالة الضرب - من الواضح انه أسرع إعادة حساب ناتج ضرب عددين بدلاً من تخزني جدول الضرب بالكامل. كيف نُقرّر؟

تحليل التكاليف والفائدة

-- يُقرّر متى المعلومات تكون متوفرة أولاً بغض النظر عن كيفية تخزينها. التحليل يُمكن أن يَرجح وزن وقيمة التخزين المطلوب، كلفة الحساب، وإمكانية الإستدعاء.

النسيان الإنتقائي

-- هنا نسمح للمعلومات لكي تُخزن مبدائياً ونُقرّر لاحقاً إذا كنا سَنحتفظ بها أو لا. بشكل واضح ان نلغوا إعادة الإستعمال إجراء جيد. نحن يُمكن أن نحدد (نعطي علامة) جسم يؤمن إستعماله الأخير. إذا كانت الذاكرة الوسيطة كاملة (ممتلئة) ونحن زُريد إضافة فقرة جديدة فإننا نُزيل الجسم الأقل إستعمالاً مؤخراً ونضيف الجديد. الاختلافات يُمكن أن تتضمن نوع من تحليل التكاليف والأرباح للتفرير إذا الجسم يجب أن يُزال.

التعلّم بأخذ النصيحة

فكرة أخذ النصيحة في نظم الذكاء الاصطناعي المبنية على اساس ال تعلّم اقترحَت بحدود 1958 (McCarthy). على أية حال القليل جداً من المحاولات التي بذلت في خلق مثل هذا النوع من الأنظمة حتى أواخر السبعينات. الأنظمة الخبيرة تُزوّد بحافز رئيسي في هذا المجال. هناك نظرتان أساسيتان في أخذ النصيحة:

- مكنته (أتمتة) كل سمات الاخذ بالنصيحة : نأخذ ال نصيحة التجريدية العالية المستوى وتُحوّلها إلى القواعد التي يُمكن أن تُوجّه عناصر أداء النظام.
- تطوي أدوات متطورة مثل محرري ومنقحي قواعد المعرفة. هذه الادوات تشتمل لمُساعدَة الخبير لترجمة خبرته إلى القواعد المُفصّلة. يعتبر الخبير هنا عنصر مكمل لنظام التعلّم. مثل هذه الأدوات مهمة في مجال الأنظمة الخبيرة من أنظمة الذكاء الاصطناعي.

أخذ النصيحة الآلي

تُلخّص الخطوات التالية هذه الطريقة:

الطلب

-- هذا يُمكن أن يُكوّن سؤالاً بسيطاً، يُسأل عن نصيحة عامّة، أو معقّد أكثر بتمييز العيوب في قاعدة المعرفة ويُسأل عن علاج.

التفسير

-- يُترجم النصيحة إلى التمثيل الداخلي.

التجهيز للعمل

-- النصيحة المترجمة قد لا تكون صالحة للاستعمال لذا في هذه المرحلة نقوم بإعادة التمثيل الذي يُمكن أن يُكوّن مستعمل من قبل عنصر الأداء.

الدمج

-- عند إضافة معرفة جديدة إلى قاعدة المعرفة يجب الأخذ بالحسبان من الآثار الجانبية السيئة لتجنّبها. ومثال على ذلك: - التكرار والتناقضات.

التقييم

-- النظام يجب أن يُقيّم المعرفة الجديدة من ناحية الأخطاء، تناقضات الخ. الخطوات يُمكن أن تُكرّر.

صيانة قاعدة المعرفة

بدلاً من أن أتمنّى الخطوات الخمس اعلاه، العديد من الباحثين جمّعوا الأدوات التي تُساعد على تطوير وصيانة قاعدة المعرفة.

العديدون منهم ركّزوا على:

- التّوّبيّ بمحررين أذكىء ولغات تمثيل مرنة لدمج المعرفة الجديدة.
 - التّوّبيّ بإدوات تنقيح لتقييم وإيجاد التناقضات والفصل في قاعدة المعرفة الحالية.
- لتمثال على هذا النظام Emycin..

مثال نظم التعلم

تعلّم لعبة الكبة (رمز القلب في ورق اللعب)

يُحاول تحويل النصيحة ذات المستوى العالي (مبادئ، مشاكل، طرق) إلى إجراءات قابلة للتنفيذ وفعّالة (LISP). القلوب:

- تلعب اللعبة كسلسلة من الحيل.
 - لاعب واحد - الذي يتصدر يبدأ اللعبة - يمي كرت.
 - اللاعبون الآخرون يلعبون تبعاً كل بدوره ويمون كرت.
 - اللاعب يجب أن يلعب مباشرة.
 - إذا هو لا يستطيع فهو يرمي أي كرت من أوراقه.
 - اللاعب الذي يمي الكرت بلقيمة الأعلى يبيح الحيلة (الدور) والتصدر.
 - يأخذ اللاعب الفائز البطاقات التي رميت في هذا الدور (الخدعة).
 - الهدف أن يتجنّب أخذ النقاط. يعتبر كل قلب نقطة واحدة، بنت البستوني تساوي 13 نقطة.
 - الفائز هو الشخص الذي بعد كل الخدع التي لعبت عنده أقل النتائج (عدد النقاط).
- الكبة هي لعبة المعلومات الجزئية بدون خوارزمية معروفة للفوز. بالرغم من أن الحالات المحتملة فأن نصائح عامة عديدة يُمكن أن تُعطي مثل:

- تجنب أخذ النقاط.
- لا تصدر بطاقة عالية في حالة ان الخصم فارغ.
- إذا الخصم لديه بنت البستوني، حاول ملاحقتها لجعله يربحها.

لأخذ نصيحة يجب على الإنسان تخطيطها إلى تمثيل مفهوم حسب لغة البرمجة المستخدمة.

```
avoid (take-points me) (trick))
```

```
(achieve (not (during
  (scenario
    (each p1 (players) (play-card p1))
    (take-trick (trick-winner)))
    (take-points me))))
```

على أية حال النصيحة ما زالت ليست شغالة حيث أنها تعتمد على نتيجة الخدعة التي هي عموماً ليست معروفة. يستعمل تحليل الحالة (أثناء التعبير) لتقرير أي الخطوات يمكن أن تُفَسَّحَ لأخذ النقاط. الخطوة 1 تُسْتثنى والخطوة 2 take-points مُتَجَلِيَّة:

```
(achieve (not (exists c1 (cards-played)
  (exists c2 (point-cards)
    (during (take (trick-winner) c1)
      (take me c2))))))
```

المحلل يجب أن يقرّر: تحت أي شروط تعمل (take me c2) تحدث أثناء (take (trick-winner) c1). التقنية، المسماة بالمجاراة (المطابقة) الجزئية، يفترض بأن النقاط ستأخذ إذا :
 نحن يمكن أن نحول تعبيرنا إلى: me = trick-winner and c2 = c1

```
(achieve (not (and (have-points (card-played))
  (= (trick-winner) me ))))
```

هذا يعني لا يتبجح الدور "خدعة" التي فيها نقاط. نحن لا نعرف من هو الفائز بللخدعة trick-winner، أيضاً نحن لم نقل أي شيء حول كيفية اللعب في الخدعة التي عندنا فيها نقطة القيادة. بعد بضعة خطوات أكثر لإنجاز هذا نحصل على :

```
(achieve (>= (and (in-suit-led (card-of me))
  (possible (trick-has-points)))
  (low (card-of me)))
```

المحلل كون من قاعدة المعرفة الأولية التالي:

- مفاهيم المجال الأساسية مثل الخدعة، يد، بدلات طابق، يتفادي، فوز الخ.
- القواعد والقيود السلوكية -- قواعد عامة عن اللعبة.
- الموجه والمساعد بالنسبة إلى كيفية التجلي.

لدينا هنا عيان أساسين:

- يفترض إلى تركيب التحكم الذي يمكن أن يطبق التشغيل الآلي.
- هو محدد إلى الكبة والمهام المماثلة.

التعلم بحل المشكلة

هناك ثلاثة طرق أساسية التي يمكن فيها النظام أن يتعلم من تجاربه الخاصة.

التعلم بتعديل العوامل

تعتمد عدة برامج على إجراء التقييم لتلخيص حالة البحث الخ. تزود برامج اللعب بالعديد من هذه الأمثلة. على أية حال، عدة برامج لها وظيفة تقييم ساكنة. في التعلم مطلوب التعديل الطفيف من صياغة تقييم المشكلة. هنا المشكلة لها وظيفة تقييم الهمثلة كمتعدد الحدود من الشكل التالي:

$$c_1 t_1 + c_2 t_2 + c_3 t_3 + \dots$$

يُعيّن t قيم الميزات و c الأوزان.

في تصميم البرامج من الصعب في أغلب الأحيان التقرير بالنسبة للقيمة المضبوطة لإعطائها لكل وزن مبدئياً. لذا الفكرة الأساسية من فكرة تعديل العوامل هي:

- تبدأ بغض التخمين حول اوضاع الوزن الصحيح.
- نعدل الوزن في البرنامج على أساس التجارب المتراكمة.
- الميزات التي نبدو كمتنبئ جيد ستضاعف أوزانها والسيدة ستقتصر.

التعلم بأوامر التشغيل المختصرة

إن الفكرة الأساسية هنا مشابهة للتعلم بالتكرار:

حيث، نتجنب إعادة الحساب المكلف. مختصرات اوامر التشغيل يمكن أن يستعملوا لتجميع مجموعة سلاسل الأعمال في واحدة، ليتم طلبها لاستعمالها لاحقاً بدلاً من طلب مجموعة كبيرة من الاوامر. على سبيل المثال: عمل العشاء يُمكن أن يُوصَف بـ : إعداد المائدة، طبخ العشاء ، مهام الخدمة. نحن يُمكن أن نُعالج "إعداد المائدة" كعمل واحد بالرغم من أنها تتضمن سلسلة من الأعمال.

إستخدم حل المشكلة بطريقة STRIPS مختصرات التشغيل في مرحلة التعليم.

لنعتبر المثال التالي ، والذي به ON(C,B) و ON(A, TABLE) حقيقة.

STRIPS يُمكن أن تُنجز ON(A,B) في أربع خطوات:

UNSTACK(C,B), PUTDOWN(C), PICKUP(A), STACK(A,B)

تُبني STRIPS الآن a مختصر تشغيل Macrop بالشروط المسبقة :

ON(C,B), ON(A, TABLE) ○

ON(A,B), ON(C, TABLE) ○

○ والخطوات الأربع كجسمها.

Macrop يُمكن أن يكون الان مستعملاً في عمليات مستقبلية. لكنه ليس عام جداً. المثال الوارد أعلاه يُمكن أن يُعمَّم بسهولة باستخدام المتغيرات بدلاً من الكتل. على أية حال التعميم ليس دائماً سهلاً.

التعلم بالقطع الكبيرة

يتضمن التعلم بالقطع الكبيرة أفكار مماثلة إلى مختصرات التشغيل ويُنشأ من الأفكار النفسية المتعلقة بحل المشكلة والذاكرة. إن القواعد الحسابية موجودة في أنظمة الإنتاج (نوقشت سابقاً). SOAR - هو النظام الذي استخدم قواعد الإنتاج لتمثيل معرفته. يستخدم القطع الكبيرة أيضاً للتعلم من التجربة.

الخلاصة الأساسية لطريقة Soar

- Soar يحل المشاكل بإطلاق المنتجات المخزونة في ذاكرة المدى البعيد.
- عندما يكتشف Soar سلسلة مفيدة من المنتجات، يقوم بتكوين القطع الكبيرة منها.
- القطع جوهرياً هي إنتاج كبير الذي يقوم بعمل سلسلة كاملة من السلاسل الأصغر.
- القطع يمكن ان تُعمَّم قبل التخزين.

التعلم الإستقرائي

هذا يتضمن عملية التعلم بالمثال -- حيث يحاول النظام البحث عن القاعدة العامة من مجموعة الحالات الملاحظة. هذا يتضمن التصنيف -- الذي ينسب، لتفصيلات (أجزاء) المدخل، اسم الصنف الذي ينتمي إليه. التصنيف مهم للعديد من مهام حل المشكلة.

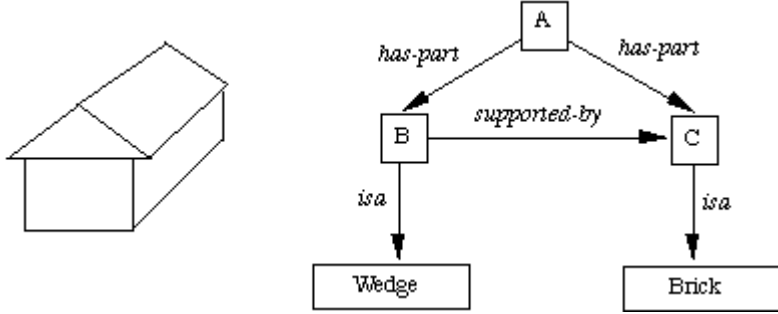
نظام التعلم يجب أن يكون قادر على تطويع أوصاف أصنافه الخاصة:

- تعاريف الصنف الأولية قد لا تكون كافية.
- العالم قد لا يكون مفهوم أو سريع التغيير بشكل جيد جداً.
- إن مهمة بناء تعاريف الصنف تُدعى تعلم المفهوم (الفكرة) أو الإستقراء

مثال على التعلم الإستقرائي -- Winston (1975)

- الهدف أن يبنى تمثيل تعاريف المفاهيم في هذا المجال.

- مفهوم مثل "البيت" يتكون من - جسم المنزل (كتلة مستطيلة) مع السقف (كتلة مثلثية) وضعت بشكل مناسب على الجسم، خيمة - حيث وتدين متلامسان جنباً إلى جنب، أو قوس - طابوقتان بلا تلامس يدعمان وتدًا ثالثاً أو الطابوق.
- فكرة أجسام رمية خاطئة قريبة -- مشابهة للحالات الفعلية التي قُدمت.
- المدخل كان خطاً يرسم تراكيب (بنية) الكتل.
- المدخلات عولجت (انظر قسم الرؤية لاحقاً) لإنتاج التمثيل اللفظي للوصف الهيكلي للجسم (الرسم. 27)



الرسم. 23 شبكة لفظية لتمثيل منزل

- الصلات في الشبكة تتضمن: يسار، يمين، لا يلامس، مدعوم بـ، له جزء، والمضمون.
 - يلامس هي علاقة مهمة - حيث جسمان مع حافة مؤثرة مشتركة يعني متلامسان.
- هناك ثلاث خطوات أساسية إلى مشكلة صياغة المفهوم:
1. اختر واحد يُعرف حالة المفهوم. إدغ هذا تعريف المفهوم.
 2. إفحص تعاريف الحالة المعروفة الأخرى للمفهوم. عمّم التعريف لتضمينهم.
 3. إفحص أوصاف التصادمات الوشيكة. حدّد التعريف لإستثناء هذه.
- كلتا خطوات 2 ، 3 تعتمدان على المقارنة وكلتا التشابهات والإختلافات من الضروري أن تُميز.

فراغات نسخة

انظمة التعلم المبنية على اساس المفهوم الهيكلي ليست بدون مشاكل. إنّ المشكلة الأكبر بأنّ المعلم يجب أن يوجّه النظام خلال السلاسل المُختارة بعناية من الأمثلة. في برنامج وينستن، طلب العملية مهم، حيث ان، صلات جديدة إضريت عند تجميع المعرفة. إنّ مفهوم أهداف فراغات النسخة غير حراس بالنسبة إلى طلب المثال. لعمل هذا بدلاً من تطوري وصف مفهوم وحيد لمجموعة الأوصاف المحتملة المُنبقعة. كأمثلة جديدة رُفدُ المجموعة التي تتطور كعملية الحالات الجديدة والتصادمات الوشيكة.

نحن سنفترض بأنّ كلّ شق في وصف فضاء النسخة يتكوّن من مجموعة المسندات التي لا تنفي مسندات أخرى في المجموعة - حرفية (موضوعية) ايجابية. في الحقيقة نحن يُمكن أن نُمثل الوصف على اساس تمثيل الإطار بعدة شقوق أو في الحقيقة إستعمال تمثيل أكثر عمومية. لأجل تبسيط المناقشة نحن سنلتزم بالتمثيل البسيط. إذا التزمنا بالتعريف أعلاه فإن خوارزمية إزالة المرشح (Mitchell's) هي أفضل خوارزمية معروفة لهذا الغرض.

لنأخذ مثال حيث انه لدينا عدد من أوراق اللعب ونحن نحتاج لمعرفة فيما إذا الكرت (ورقة اللعب) مفرد وأسود.

مبدائياً نحن نعرف أشياء مثل: الأحمر، الأسود، البستوني، السباتي، كرت زوجي، كرت فردي.. الخ. لذا الطاقة 8♥ هي حمراء، وزوجية وكبة.

الخصائص الاساسية لتحديد طريقة فضاء النسخة:

- مفاهيم ربط في المجال يُمكن أن تُطلب جزئياً بالتحديد.
- في مثال ورق اللعب، مفهوم "الأسود" أقل تحديداً من مفهوم "الأسود المفرد" أو "الاسود المفرد البستوني".
- المفاهيم "أسود - مفرد" و "بستوني" غير قابلة للمضاهاة، حيث لا يوجد تعيين أكثر (أو أقل).

- مفهوم "أسود" أكثر تعييناً من "أي كرت".
- مجموعة التدريب تشمل مجموعة من البطاقات ولكل كرت يحدد فيما إذا كان في مجموعة الهدف (أسود مفرد) أو لا.
- إن مجموعة التدريب عولمت بشكل تزايدى وقائمة المفاهيم المعيّنة "الأكثر والأقل" متسقة مع تدريب الحالات المتبقية.
- لنرى كيف يمكن التعلّم من مجموعة عيّرات مدخلة:
- مبدائي المفهوم الأكثر تعييناً والمتسق مع البيانات هو المجموعة الفارغة. المفهوم الأقل تعييناً هو مجموعة كل البطاقات.
- دغ $4\spadesuit$ تكون البطاقة الأولى في مجموعة العيّنة. هذه البطاقة أسود مفرد.
- لذا المفهوم الأكثر تعييناً $4\spadesuit$ لوحده ما زال كل مالدينا من بطاقات.
- البطاقة التالية: $9\clubsuit$ نحتاج لتعديل مفهومنا المعين أكثر للإشارة إلى تعميم المجموعة، شئ مثل: "بطاقات مفردة وسوداء".
- البطاقة التالية: $4\heartsuit$ الآن نحن يمكن أن نعدل المجموعة أقل تعييناً لإستثناء $4\heartsuit$. حيث ان الإستثناءات الأكثر مضافاً فلننا سنعمّم هذا إلى كل البطاقات السوداء وكل البطاقات المفردة.
- لاحظ ان الحالات السلبية تُسبب التالي: أقل المفاهيم المعيّنة تُصبح أكثر تعيّن، والأكثر إيجابية تُؤثّر على المعيّنة الأكثر بنفس الطريقة.
- إذا تُصبح المجموعتان نفس المجموعة فإن النتيجة مُضمّنة ومفهوم الهدف يُحقّق.

خوارزمية إزالة المرشّح

لنصف الخوارزمية الآن.

لنفترض G هي مجموعة أكثر المفاهيم العامة. ولنفترض S هي مجموعة أكثر المفاهيم المعيّنة. افترض: عندنا لغة تمثيل مشتركة ولدينا مجموعة أمثلة التدريب السلبية والإيجابية. الهدف: وصف المفهوم الذي يتسق مع كل الإيجابية فقط من الأمثلة السلبية. الخوارزمية:

- عرف G لإحتواء عنصر واحد -- الوصف الملغي، كل الميزات متغيّرات.
- عرف S لإحتواء عنصر واحد للذي في المثال الإيجابي الأول.
- إعادة (كر)

0 أدخل مثال التدريب التالي

0 إذا كان مثال إيجابي -- لخطوة أولى احذف من G الأوصاف التي لا تغطي المثال. ثم جدد S لإحتواء المجموعة الأكثر تعييناً من الأوصاف في النسخة التي تغطي المثال ومجموعة العنصر الحالية في S . وبمعنى آخر: . عمّم عناصر S لتكون صغيرة بقدر الإمكان لكي يغطون مثال التدريب الجديد.

0 إذا كان مثال سلبي -- لخطوة أولى احذف من S الأوصاف التي تغطي المثال. ثم جدد G لإحتواء المجموعة الأكثر عمومية من الأوصاف في النسخة التي لا تغطي المثال. وبمعنى آخر: . تخصّص عناصر S صغيرة بقدر الإمكان لكي لا تغطي الأمثلة السلبية بعناصر G .

حتى S و G كلنهما يصبحا مجموعات الورقة الوحيدة (شرط التكرار).

- إذا S و G كانتا متطابقتان فإننا نعطي قيمة نتائجهم.

- إذا S و G مختلفتان، فإن مجموعات التدريب متناقضة.

لننظر الان إلى مشكلة تعلّم المفهوم "الفلوش" في لعبة البوكر، حيث كل خمس بطاقات من نفس النوعية.

($5\clubsuit, 7\clubsuit, 8\clubsuit, J\clubsuit, K\clubsuit$)

لنجعل المثال الأول يكون إيجابياً:

ثم وضعنا

$$G = \{(x_1, x_2, x_3, x_4, x_5)\}$$

$$S = \{(5\clubsuit, 7\clubsuit, 8\clubsuit, J\clubsuit, K\clubsuit)\}$$

(5♣, 5♥, 6♦, J♥, A♠)
 المثال الثاني سلبي:
 نحن يجب أن نخصص G (فقط إلى المجموعة الحالية):

$$G = \{(x_1, x_2 = 7\clubsuit, x_3, x_4, x_5),$$

$$(x_1, x_2, x_3 = 8\clubsuit, x_4, x_5),$$

$$(x_1, x_2, x_3, x_4 = J\clubsuit, x_5),$$

$$(x_1, x_2, x_3, x_4, x_5 = K\clubsuit)\}$$

S غير متأثر.
 مثالنا الثالث إيجابي:

تزيل التضاربات أولاً من G وبعد ذلك نعمم S:

$$G = \{(x_1, x_2 = \clubsuit, x_3, x_4, x_5),$$

$$(x_1, x_2, x_3 = \clubsuit, x_4, x_5),$$

$$(x_1, x_2, x_3, x_4 = \clubsuit, x_5),$$

$$(x_1, x_2, x_3, x_4, x_5 = \clubsuit)\}$$

$$S = \{(x_1 = 5\clubsuit, x_2 = \clubsuit, x_3 = \clubsuit, x_4 = \clubsuit, x_5 = \clubsuit)\}$$

مثالنا الرابع إيجابي أيضاً:

تزيل التضاربات مرة أخرى من G وبعد ذلك نعمم S:

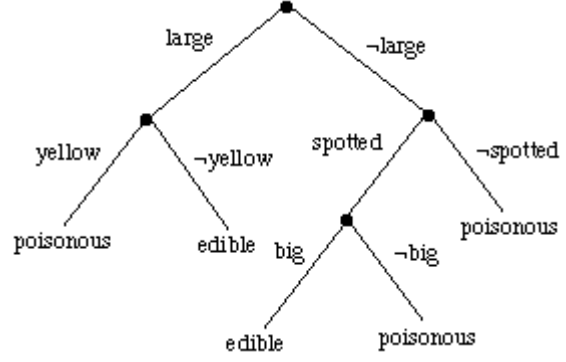
$$G = \{(x_1 = x_2 = x_3 = x_4 = x_5 = \text{same suit}, x_2, x_3, x_4, x_5), \}$$

$$S = \{(x_1 = x_2 = x_3 = x_4 = x_5 = \text{same suit}, x_2, x_3, x_4, x_5)\}$$

- نحن يمكن أن نستمر بالتعميم والتخصيص
- أخذنا بضعة قفزات كبيرة في مخطط التخصيص / التعميم في هذا المثال. تطلب المزيد من خطوات التدريب للوصول الى هذه الخاتمة.
- من الصعب إكتشاف اتجاه النوع نفسه ... الخ.

أشجار القرار

Quinlan قدم فكرة أشجار القرار في نظامه ID 3 ، البرنامج الذي يمكن أن يبني أشجار القرار آلياً من الحالات الإيجابية والسلبية المعطية. أساساً كل ورقة شجرة قرار تُصرّح مفهوم إيجابي أو سلبي. لتصنيف مدخل معين نبدأ في القمة وننزل المزاعم أسفل فأسفل حتى نصل الجواب (الرسم 23)



الرسم. 23 شجرة قرار صالحة فطر للأكل

بناء أشجار القرار

- استعمل ID3 الطريقة التكرارية.
- الأشجار البسيطة فضلت كتصنيف أكثر دقة.
- الاختيار العشوائي من عينات تدريب المجموعة تم للتجميع الأولي من الشجرة -- المجموعة الثانوية للنافذة.
- أمثلة تدريب أخرى تُستعمل لإختبار الشجرة.
- إذا صُنفت كل الأمثلة بشكل صحيح توقف.
- بخلاف ذلك اضع عدد من أمثلة التدريب إلى النافذة وابدأ ثانية.

إضافة عقدة جديدة

- عند نجمع الشجرة نحتاج لإختبار وقت إضافة العقدة الجديدة:
- بعض الخواص ستنتج معلومات أكثر من الآخرين.
- إضافة عقدة جديدة قد تكون عديمة الفائدة في عملية التصنيف العامة.
- أحياناً الخواص ستفصل حالات التدريب إلى مجموعات فرعية و التي أعضائها يشتركون في علامة مشتركة. هنا نقرر يمكنه إنهاء والعقدة الورقة خصصت للمجموعة الثانوية الكاملة.

فوائد شجرة القرار:

- أسرع من فراغات النسخة إذا فضاء المفهوم كبير.
- الملتقى أسهل.

الأضرار:

التمثيل ليس طبيعي بالنسبة للبشر -- قد يكون من الصعوبة توضيح تصنيف شجرة القرار.

التعلم على اساس التفسير (EBL)

- يستطيع البشر تعلم الكثير من مثال واحد.
- الفكرة الأساسية: استعمال نتائج مشكلة ما يوفر جهداً في المرة القادمة.
- التعلم على اساس التفسير يقبل 4 أنواع المدخلات:

مثال التدريب

- حيث يتعلم من ما يراه في العالم.

مفهوم الهدف

- مستوى الوصف العالي الذي يفترض ان يتعلمه البرنامج.

معيار جاهزية العمل

- وصف المفاهيم القابلة للإستعمال.

نظرية المجال

- مجموعة القواعد التي تصف العلاقات بين الأجسام والأعمال في مجال ما.

التعلم المبني على التفسير يَحْسَبُ تعميم مثال التدريب الذي لَيْسَ فقط كافي لوصف مفهوم الهدف لكن ليحقق معيار جاهزية العمل أيضاً.
لعمل هذا هناك خطوتان:

التفسير

-- نظرية المجال تُستعمل لتَشْدِيب كُلِّ السمات الغير مهمة مِنْ مثال التدريب فيما يتعلق بمفهوم الهدف.

التعميم

-- التفسير مُعَمَّم كـمَحْتَمَل بعيد بينما ما زال يَصِفُ مفهوم الهدف.

مثال على التعلم المبني على التفسير

الهدف: الوصول إلى حمص (Homs) -- مدينة سورية.
إنَّ بيانات التدريب:

```
near(Damascus, Homs),
airport(Damascus)
```

إنَّ معرفة المجال هي:

```
near(x,y) ^ holds(loc(x),s) → holds(loc(y), result(drive(x,y),s))

airport(z) → loc(z), result(fly(z),s))
```

في هذه الحالة معيار جاهزية العمل : نحن يَجِبُ أَنْ نَبْدِي تعريف المفهوم بوصف نقي من اللغة.
هدفنا يُمكن أَنْ يَبْدُو كالتالي:

holds(loc(Homs),s) -- جِدْ بَعْض الحالات $s \in \text{holds}$.

نحن يُمكن أَنْ نُنَبِّت ان holds مع s مُعرَّف مِنْ قَبْل:

```
result(drive(Damascus,Homs),
result(fly(Damascus), s'))
```

نحن يُمكن أَنْ نَطِيرَ إلى مطار دمشق (Damascus) وبعد ذلك نَقُودُ بالسيارة إلى حمص Homs. إذا حلَّلنا البرهان يُمكن أَنْ نَتَعَلَّمَ بضعة قواعد عامَّة منه. حيث ان حمص Homs تَظْهَرُ في الإستفسار نحن يُمكن أَنْ نُجَرِّدَ لإعطاء التالي:

```
holds(loc(x), drive(Damascus,x),
result(fly(Damascus), s'))
```

لكن هذا ليس صحيحا بالضبط – حيث اننا لا نَسْتَطِيعُ الوصول الى كل مكان بلطَيْرَان إلى دمشق. حيث Homs يَظْهَرُ في قاعدة البيانات عندما نُجَرِّدُ الأشياء، نحن يَجِبُ أَنْ نُسَجِّلَ إستعمال الحقائق بشكل واضح:

```
near(Damascus,x) → holds(loc(x), drive(Damascus,x), result(fly(Damascus),
s'))
```

هذا يَذْكُرُ إذا x قُرْب دمشق نحن يُمكن أَنْ نَصِلَ إليه بلطَيْرَان إلى دمشق وبعد ذلك قيادة السيارة الى x. اذن نحن نَعْلَمُنا هذه القاعدة العامة.

نحن يُمكن ايضا أَنْ نُجَرِّد دمشق بدلاً مِنْ Homs لَنُصَبِّحَ:

```
near(Homs,x) ^ airport(x) → holds(loc(Homs), result(drive(x,Homs),
result(fly(x), s')))
```

هذا يَصْرَحُ بأننا يُمكن أَنْ نَصِلَ الى حمص Homs بلطَيْرَان إلى اقرب مطار وقيادة السيارة مِنْ هناك. نحن يُمكن أَنْ نُضَيِّفَ مطارَ (حلب (Aleppo)) وَنَحْصِلُ على وسائل بديلة مِنْ خطَّة السفر. أخيراً نحن يُمكن أَنْ نُجَرِّدَ في الحقيقة كل من Homs و Damascus لَنُصَبِّحَ الخطة العامة:

holds (loc (y) , \rightarrow airport (y) \wedge near (x, y)
result (drive (x, y) , result (fly (x) , s'))

الإكتشاف

الإكتشاف شكلٌ مقيد من العَلَم الذي فيه يكتسبُ الكيانُ الواحد المعرفةً بدون مساعدةٍ معلّم.

نظريةُ الإكتشاف المقاد - (1976)

AM - هو برنامجٌ يكتشفُ المفاهيم في الرياضياتِ الأولية ونظرية المجموعات.

AM يملك مُدخلان:

- وصف بعض مفاهيم نظرية المجموعات (في شكلٍ LISP). ومثال على ذلك: - إتحاد المجموعات ، التقاطع، المجموعة الفارغة.
- معلومات عن كيفية انجاز الواضيات. ومثال على ذلك: - العمليات في الرياضيات.

بإعطاء المعلومات أعلاه AM إكتشفت:

الأعداد الصحيحة

-- من المحتمل حساب عناصر المجموعة وهذه الصورة هي وظيفة الحساب -- الأعداد الصحيحة - مجموعة مثيرة بسبب ملائمتها.

الإضافة

-- إتحاد مجموعتين مفككتين ووظيفة الحساب الخاصة بهما.

الضرب

-- بعد إكتشاف الإضافة والضرب كعمليات نظرية مجموعة مرهقة، أوصاف أكثر فعالية، كانت مزودة بلليد.

الأرقام الصماء

-- اكتشف قيمة العامل للاقام والأعداد مع عامل واحد فقط.

تخمين Golbach

-- أعداد زوجية يُمكن أن تُكتب كنتيجة جمع عددين اصمين. ومثال على ذلك: - $11 + 17 = 28$.

أعداد قابلة للقسمة بشكل أعلى

-- أعداد مع عوامل أكثر قدر الامكان. أي عدد k يكون قابل للقسمة بشكل أعلى، اذا كان k عنده عوامل أكثر من أي

عدد صحيح أقل من k. ومثال على ذلك: - 12 عنده ستة مقسومات 1,2,3,4,6,12.

كيف يعمل AM ؟

AM يستخدم العديد من التقنيات العامة للذكاء الاصطناعي:

- التمثيل المبني على اساس الأطر للمفاهيم الرياضية.
- AM يُمكن أن يخلق مفاهيم جديدة (شقوق) ويملأ قيمهم.
- استخدم البحث الإرشادي
- الموجه المساعد 250 يُمثل التلميحات حول النشاطات التي قد تؤدي إلى إكتشافات مهمة.
- كيف يستخدم الوظائف، يخلق مفاهيم جديدة، تعميم الخ.
- بحث مبني على الفرضية والاختبار .
- جدول أعمال (اجندة) لإكتشاف مراحل العمل.

إكتشاف البيانات المقاد -- سيكون (1981)

العديد من الإكتشافات تمت بملاحظة البيانات من العالم وفهمها -- ومثال على ذلك: - الفيزياء الفلكية - إكتشاف

الكواكب، ميكانيك الكم - إكتشاف الجزيئات الذرية الفرعية.

يبكون هو محاولة لتنفيذ مثل هذه النظم من الذكاء الاصطناعي.

خلاصة نظام بيكون:

- ابدأ مع مجموعة المتغيرات للمشكلة.

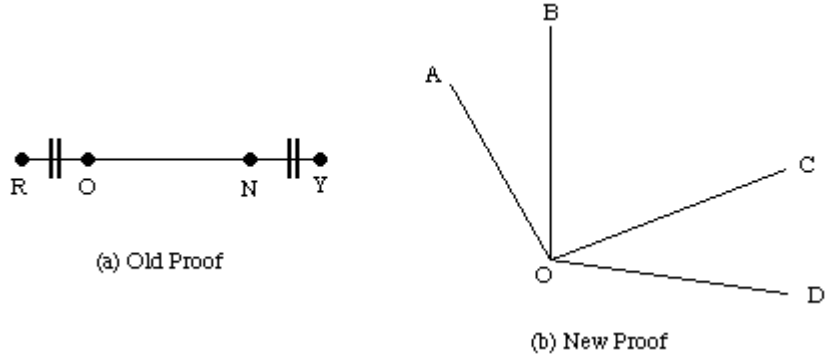
- ومثال على ذلك: - سيكون كَانَ قادرَ على إشتقاق قانون الغاز المثالي. بدأ بأربعة متغيرات p - ضغط الغاز، V -- حجم الغاز، n -- كتلة من الغاز، T -- درجة حرارة الغاز. الإستدعاء - $pV/nT = k$ ، حيث أن k ثابت.
- ادخال قِيم من البيانات التجريبية من المشكلة.
- سيكون يَحْمِلُ بَعْض الثوابت ويُحاول مَلاحَظَة الإِتِّجَاهَات في البيانات.
- إعطاء الإستنتاج.
- سيكون أيضاً طبق في قانون Kepler الثالث، قانون Ohm ، حماية الزخم وقانون Joule.

التناظر

التناظر يتضمّن التخطيط المعقد بين المفهومين الذان يبدوان متباينان. زيد يُبْنَى غرفة كبيرة من الطابوق خارج المنزل. هو كَانَ مثل المعجون في أيديها. يميز البشرُ بسرعة الفكرة التجريدية المتضمنة ويفهموا المعنى. هناك طريقتان من طرق المشاكلة القياسية درسا في الذكاء الاصطناعي.

التناظر التحويلي

إبحث عن حَلّ مماثل وإنسخه إلى الحالة الجديدة التي تجعلُ البدائل المناسبة حل ملائم. ومثال على ذلك: - الهندسة الفراغية. إذا كنا نَعْرِفُ معلومات حول أطوال قطع الخطوط و الزوايا بأن بَعْض الخطوط متساوية (الرسم. 24) فإننا يُمكنُ أن نَضَعُ فرضيات مماثلة بالنسبة للزوايا.



الرسم. 24 مثال تناظر تحويلي

- نَعْرِفُ بأن الخطوط $RO = NY$ والزوايا $AOB = COD$
- رَأَيْنَا ان $RO + ON = ON + NY$ - قاعدة مضافي
- لذا نحن يُمكنُ أن نَقُولُ بأن الزوايا $AOB + BOC = BOC + COD$
- بناءا على قاعدة التعدي فإن الخط $RN = OY$
- لذا بنفس الطريقة فإن الزاوية $AOC = BOD$
- يَصِفُ Carbonell (1983) T -space ، طريقة الفضاء T لتحويل الحلول القديمة إلى جديدة.
- كَلّ الحلول مَنْظُورَة لحالات في فضاء المشكلة - فضاء T .
- مشغلي T يَصِفُونَ طرقَ تحويل حالات الحَلّ الناتج إلى الحالات الجديدة.
- التفكّي بالتناظر يُصَبِحُ بحث في فضاء T -- تحليل وسائل الطرفيات.

التناظر المشتق

التناظر التحويلي لا يَنْظُرُ إلى لثيفية حل المشكلة -- ولكن يَنْظُرُ إلى الحَلّ النهائي فقط.

إنَّ تأريخَ (سلسلة الخطوات) حلَّ المشكلة – الخطوات المأخوذة والمنفذة من البداية وحت الوصول الى الحل - في أغلب الأحيان ذات صلة كبيرة.

- أظهر Carbonell بأنَّ التناظر المشتق مكوّن ضروري في عملية نقل المهارات في المجالات المعقّدة:
- في ترجمة رموز لغة البرمجة باسكال الى لغة البرمجة LISP – الترجمة الحرفية خطّ بخطّ لا تفيد. لكن يجب أن نستعمل الهيكلية الرئيسية وقرارات التحكم.
 - طريق واحد لعمل هذا هو أن نعيد الاشتقاق السابق ونعدّله عند الضرورة.
 - إذا الخطوات والفرضيات الأولية ما زالا نسخة صحيحة فأنا نعبرهم.
 - بخلاف ذلك نحتاج لإيجاد بدائل -- أفضل بحث أولى.

الإدراك العام

تبدي الأنظمة الذكية الحقيقية إدراك عام – هذه الانظمة تم تلك معرفة أكثر من اللازم لكي تكون قادرة على العمل في البيئة المعطاة. ذكرنا نظام CYC الذي يعتبر محاولة طموحة لتشفير الحس العام. على أية حال يظهر هذا المثال كيف ان هذا النوع من النظم تتطلب قاعدة معرفة كبيرة جداً. نحتاج أنظمة الحس العام لدعم:

- أوصاف الأشياء (الأجسام) العادية -- الإطار.
 - سلاسل مثالية من الأحداث اليومية – مخطوطات (برمجيات).
 - استنتاج (تفكير) التركيبية -- المنطق الرتيب.
- تُصوّر إستراتيجيات الحس العام العديد من المواضيع المهمة في الذكاء الاصطناعي. نحن سنناقش كيف هذا يُمكن أن يُطبّق على العديد من المواضيع التي نوقشت سابقاً.

العالم الطبيعي -- الفيزياء النوعية

الفيزياء النوعية هي احدى مناطق تطبيق أنظمة الذكاء الاصطناعي المهتمة بالتفكّي حول سلوك الأنظمة الطبيعية (الفيزيائية). هي منطقة جيدة للدراسة حيث أن الإنسان يعرف معلومات عظيمة حول هذا العالم، على سبيل المثال:

- الإنسان يُمكن أن يتوقع انه عند سقوط الكرة ستقفز في العديد من الحالات.
 - هو يُمكن أن يتوقع مكان سقوط (مسقط) كرة الكريكت والإمساك بها حتى.
 - يعرف ان الهدول (النواس) سيهتقر في المنتصف بعد تأرجحه للأمام والخلف.
- على أية حال أكثر البشر بينما هم يؤدون اعمالهم في هذا العالم ليس لديهم فكرة كبيرة عن قوانين الفيزياء التي تحكم هذا العالم. نحن يُمكن أن ننظر الى المعلومات بشكل واضح ونشتق المعادلات لوصف، على سبيل المثال حركة الهدول. في الحقيقة الحاسبات جيدة جداً في هذا النوع من الحساب بتوفر بُرامج مصممة من قبل المبرمجين الخبراء. هل هذه هي الكيفية التي يعمل بها النظام الذكي في هذا العالم؟
- الأطفال بعمر الثلاثة سنوات هل بإمكانهم أن يقرأوا أو يحلوا رياضيات أولية؟
- حافز آخر وهو بوجود نماذج حاسوبية معقدة يُمكن أن نُجمع العديد من المشاكل التي تعتبر صعبة أو مستحيلة الحل بشكل تحليلي. أنظمة المعادلات (التفاضل، التكامل... الخ) قد تكون صعبة لأن تشتق ومستحيلة حتى للحل.

نمذجة العالم النوعي

الفيزياء النوعية تسعى لفهم العمليات الطبيعية ببناء النماذج منهم. أي نموذج قد يشمل الكيانات التالية:

المتغيرات

-- تاخذ القيم كما في نموذج الفيزياء التقليدي لكن مع مجموعة محددة من القيم، ومثال على ذلك: - درجة الحرارة :

{frozen, between, boiling}

{التجمد، بين التجمد والغليان، الغليان}

فراغات الكمية

-- مجموعة صغيرة من القيم الرصينة لمتغير.

نسبة التغير

-- المتغيرات تأخذ قيم مختلفة في الأوقات المختلفة. النسبة الحقيقية المقيمة للتغير يُمكن أن تُشكّل بشكل نوعي مع

{decreasing, steady, increasing}

فضاء الكمية، ومثال على ذلك: -. { متزايدة، ثابتة، متناقصة }

التعابير

-- مزيج من المتغيرات.
المعادلات
-- مهمة التعبير إلى المتغيرات.
الحالات
-- مجموعات المتغيرات التي تغير قيمها مع الوقت.

ملاحظة : الجبر النوعي مختلف:
 $\{empty, between, full\}$ { فارغة، نصف مملوءة، مملوءة }
لنفترض اننا نصف الكاسة
ثم عندما نُضيف قيمتين نوعيتين بأن واحد نحصل على:
 $empty + empty = empty$
 $empty + between = between$
 $empty + full = full$
 $between + between = \{between, full, overflow\}$
 $between + full = full + overflow$
 $full + full = full + overflow$

التفكير (الإستنتاج) مع المعلومات النوعية
التفكّسي في هذه المنطقة يُعنى محاكاة نوعية في أغلب الأحيان. الفكرة الأساسية تصبح:

- تركيب سلسلة الحوادث المنفصلة التي تحدث كتغييرات قيمة متغيرات نوعية.
- الحالات المُرتبطة بالقواعد النوعية هي التي قد تكون عامة.
- القواعد قد تُقدّم إلى العديد من الأجسام بشكل آني ، حيث انهم يُثرون على بعضهم البعض – مستخدم هنا تحقيق القيد.
- الغموض قد يُظهر وبالتالي نفس النتائج لطرق واشكال مختلفة من شبكة كل الحالات والانتقالات المحتملة.
- كل طريق يُدعى تاريخ الشبكة هو تخيلي (تصوري).
- الأنجازَ برامج فعالة لهذا الغرض يجب أن نعرف كيف نُمثل سلوك العديد من أنواع العمليات، المواد والعالم الذي فيه يتصرفون.

الحس العام الوجودي
بعض المفاهيم أساسية بالنسبة لتفكّسي الحس العام ومنها: الزمن، المكان، المادة.
الزمن
هنا نحن نُعنون أفكار الزمن المألوفة إلى أكثر الناس كمعارض للطبيعة الفلسفية للزمن. على سبيل المثال:

- سجّل الموسيقى X ألبومات بين منتصف الستينات و1970.
- مات X في 1970.
- أصدر ناس مهتمين بالموسيقا ألبوماً مستندا على عينات كل موسيقى X المسجلة.
- نحن يُمكن أن نستنتج بسهولة بأن الألبوم المذكور اخيرا أُصدر بعد 1970.

 إنّ الفكرة الأكثر أساسية من الزمن محتلة بالأحداث:

- يقع الأحداث أثناء فترات -- أوقات زمنية مستمرة.
- الفترة لها نقاط بداية ونهاية و مدة (من الزمن) بينهم.
- الفترات يُمكن أن تُتعلق بأحدهما الأخرى -- أوصاف مثل قبل ذلك، بعد، اجتماعات، يُقابل قبل، بدايات، يُبدأ قبل، أثناء، نهايات، مُنتهى ب..الخ.
- نحن يُمكن أن نُبني الדיهييات مع الفترات لوصف الأحداث بمرور الوقت.

المكان

إنَّ عالمَ الكُتَلِ مثال بسيط حيث اننا يُمكنُ أَنْ نُشكِّلَ وَنُصِفَ المكان. على أية حال أفكار الحسِّ العام مثل "مكان الجسم x بقُوبِ جسم y " غير ملائمة. الاجسام لها مدى مكاني بينما الأحداث لها مدى مؤقت (زمني). لذا نحن قَدْ نُحاولُ توسيع نظرية الحسِّ العام مِنْ الوقت. على أية حال المكان هو فراغ ثلاثي الأبعاد 3D وهناك المزيد مِنْ العلاقات مِنْ هذه للوقت لذا هو ليس فكرة جيدة. النظريَّة الأخرى هي إظهار الاجسام والمكان في المستويات المُختلفة مِنْ التجريد. ومثال على ذلك: - نحن يُمكنُ أَنْ نُنظُرَ الى البطاقات الإلكترونية المطبوعة كما لو أنَّها جسم ثنائي الأبعاد 2D. إختيار التمثيل يعني تحديد الخصائص ذات العلاقة في المستويات المعينة مِنْ ال جزئيَّة. على سبيل المثال نحن يُمكنُ أَنْ نُعرِّف العلاقات على المكان مثل داخل، مجاورة ... الخ. نحن يُمكنُ أَنْ نُعرِّف ايضا العلاقات للأقواس وخطوط وسطوح والمستويات والحجوم. ومثال على ذلك: - على طول، عبر، عمودي الخ.

المادة

نحتاج لوصف خصائص المواد:

- أنت لا تستطيع المشي على الماء.
- إذا انت ترمي لكأس القهوة بعيدا، ماذا يحدث؟
- إذا انت تصب مغلاة كاملة في كأس ماذا يحدث؟
- أنت يُمكنُ أَنْ تُعصر الإسفنج لكن ليس الطوب.
- السوائل (كما يبدو اعلاه) تُزودُ بالعديد مِنْ النقاط المثيرة. هذا مهم للتفكير بالأمثلة المشغولة بالاجسام. هكذا نحن يُمكنُ أَنْ نُعرِّف الخصائص مثل:
- السعة -- تشير إلى كمية السائل.
- الكمية -- الحجم المشغول مِنْ قِبل السائل.
- كامل -- إذا السعة تساوي الكمية.
- خصائص المواد يُمكنُ أَنْ تتضمن جماعات:
- حر -- إذا المكان لا يُحتوي كليا اي جسم داخله.
- محيط -- إذا كان المكان محاط بفراغ حر رقيق جداً.
- متصلب
- مرن
- جزئي -- ومثال على ذلك: - الرمل

تنظيم الذاكرة

- الذاكرة تعتبر مركز بالنسبة إلى سلوك الحسِّ العام وأيضاً القاعدة الأساسية للتعلم. الذاكرة الإنسانية ما زالت ليست مفهومة بالكامل على أية حال العلماء النفسانيون إقترحوا عدة أفكار:
- "ذاكرة الهمد القريب (STM) -- فقط عدة بنود يمكن ان تحمل هنا في نفس الوقت. المعلومات الخاصة بقوة الادراك تخزن مباشرة هنا.
 - ذاكرة الهمد البعيد (LTM) -- قدرة على التخزين كبيرة جداً ودائمة.
 - ذاكرة المدى البعيد تُقسم في أغلب الأحيان الى:
 - ذاكرة إستطراذية -- تحتوي على المعلومات الخاصة بالتجارب الشخصية.
 - ذاكرة سيمانتقية (لفظية) -- تحتوي على الحقائق العامة بدون معنى شخصي. ومثال على ذلك: - الطيور تطير. وذلك مفيد في عملية فهم اللغة الطبيعي.
 - من ناحية البحث في أنظمة الذكاء الاصطناعي فإنها بدأت مِنْ قِبل Quillian على الذاكرة السيمانتقية و أدت إلى الشبكات والإطر السيمانتقية والتراكيب الأخرى لكاشق والحشوة. بالعمل على الذاكرة الإستطراذية نتجت ملخصات المخطوطات. كذلك أنظمة إنتاج هي مثال على نماذج الحاسوب STM و LTM .

دور الذاكرة في حل المشكلة

لتهيي هذا الموضوع بمعرفة دور الذاكرة في حل المشكلة. رأينا بأن العديد مِنْ المشاكل محلولة بالتناظر. أنظمة الحاسوب التي تُؤدي هذه المهمة تسمى أنظمة الإستنتاج المبنية على ال حالة (CBR). تستخدم هذه الأنظمة مكتبات حالة عامة كبيرة بدلاً مِنْ أوصاف مِنْ المبادئ الأولية. كذلك تعتمد هذه الأنظمة و بشدة على تنظيم وإسترجاع الذاكرة.

- نظام الفهرسة الغني يجب أن يُستخدم -- عندما يستنتج مشكلة ما فإن الفجارب الماضية و ذات العلاقة فقط يجب أن تذكر (تطلب من الذاكرة).
- دليل بالميزات (الخصائص) الموجودة في المشكلة.
- يتطلب نوع من معايير صلة المعلومات المُسترجعة.
- بعض الميزات المهمة فقط في السياق المؤكد.
- التعلم المبني على الاستقراء والتفسير مناسب هنا.
- تراكيب البيانات المستعملة ستكون مهمة حيث ان عدد الحالات المثلثة سيكون كبير.
- هل نسترجع كل المعلومات حول الحالة أو جزء منها؟
- عدد من الحالات تسترجع عادة. نحتاج لإختيار أفضل حالة باستخدام موجهين إرشاديين والذي قد يتضمنون:
- هدف بافضلية اولية-- الحالات التي تتضمن الهدف نفسه لئلا مشكلة الحالية.
- تفضيل ميزة بارزة -- الحالات التي تتضمن الأكثر أهمية (أو العدد الأكبر) من الميزات.
- تفضيل محدد -- مطابقة حسب الميزات المحددة.
- تفضيل ثنائي -- بتحديد الحالات المطابقة بكثرة.
- تفضيل الحديث -- بانتقاء الحالات المطابقة مؤخراً.
- تفضيل التكيف -- الحالات التي عدلت ميزاتها بسهولة للتكيف مع المشكلة الجديدة.

الرؤية

الرؤية جزء مهم من أنظمة الذكاء الاصطناعي ، وهي اعطاء خاصية البصر لهذه الانظمة و جعلها تفهم ماترى .
الرؤية هي الحاسة الأكثر تعقيدا لدى الإنسان – حيث كرس لها حول رُبع حجم الدماغ. كذلك حوالي 80% من مدخلات المعرفة التي تصل الى المخ البشري تدخل عن طريق النظر ، والـ 20% الباقية هي نصيب باقي الحواس .

تحدي الرؤية

أنظمة الرؤية المطورة حتى الآن لا تملك القدرة الموجودة لدى البشر من ناحية فهم المشاهد الداخلية أو الخارجية المعقدة. تحتاج أنظمة الرؤية الآلية الناجحة أن تكون موجهة في بيئة ضيقة إلى حد كبير.
لماذا الرؤية صعبة؟

- العالم : ثلاثي الأبعاد. الصور المشكلة في آلة تصوير هي ثنائية الأبعاد -- بعض المعلومات فُقدت و من الصعب استعادتها.
- الصورة معتمدة على العديد من العوامل:
- طبيعة أداة الإحساس – الأشعة تحت الحمراء ، تدرجات الرمادي (الصور الابيض/السود) ، اللون ... الخ .
- خصائص الأداة -- الحساسية، الوضوح، العدسات، الاستقرار، اليؤرة... الخ.
- إضاءة المشهد -- الإضاءة الخافتة تعطي مقارنة سيئة، الظلال أو الإنعكاسات المفرطة قد يسببان مشاكل أيضاً.
- البيئة -- غبار، ضباب، رطوبة... الخ.
- الخصائص العاكسة لسطح الجسم – مادة الشيء، اللون .. الخ.
- حجم كبير من المعلومات يجب أن يُعالج. حيث ان الخوارزميات المتعلقة بمعالجة الصور (الرؤية) وتمييزها تتطلب حسابات عالية الأداء وكثيرة جداً.
- معرفة كثيرة عن العالم – عن كل من الأجسام والبيئة - من الضروري أن تُمثلا لخوارزميات الرؤية.
- أي نوع من العمليات نتجها نظم معالجة وتمييز الصور لإدائها ولماذا هذا مرتبط بأنظمة الذكاء الاصطناعي؟
- وصف الصور ، والأجسام والعالم الطبيعي -- بشكل واضح نحتاج الى نماذج (رياضية) للصورة والأجسام ونحتاج الى مخطط تمثيل المعرفة أيضاً.
- معالجة الصور -- نحسن مواصفات الصور للتعامل معها بشكل افضل من قبل الحاسوب أو الإنسان. مثال على ذلك : تحديد الحواف، تعديل الاضاءة ، تنقية وتصفية الصور باستخدام المرشحات... الخ.

- التفسير - استخلاص ميزات الصورة (دلائل التمييز)، مثل الحافة، المناطق، السطوح وأوصاف أخرى من الصورة.
 - تمييز النمط -- لصور الجسم الواحد، نصنّف الصور بمطابقة القيم الدالة عليها مع القيم الدالة على باقي الصور في قاعدة البيانات أو القيم المحددة للانتماء لهذا الصنف.
 - تحليل المقياس - (معايير المطابقة) ميزات القياس في الجسم.
 - فهم الصورة - لصور متعددة الاجسام، تُحدّد مكان الأجسام في الصورة، نُصنّفهم وربما يبيّن نمودج ثلاثي الأبعاد من المشهد.
- إنّ الهدف النهائي من نظام الرؤية هو التعرف (تمييز الاجسام) على الأجسام في الصورة. من الواضح ان الكثير من تقنيات أنظمة الذكاء الاصطناعي -- تمثيل المعرفة، التفكير والاستنتاج، الفهم، التخطيط والتعلم -- مطلوبة في عملية الرؤية الآلية.

إكتساب الصورة

إنّ المرحلة الأولى لأي نظام رؤية هي مرحلة إكتساب (إدخال) الصورة. كيفية الحصول على الصورة وإدخالها الى الحاسوب. بعد الحصول على الصورة، هناك طرق مختلفة من المعالجة يُمكن أن تنفذ على الصورة لإداء العديد من مهام الرؤية المختلفة والمطلوبة اليوم. على أية حال، إذا الصورة لم تُكتسب بشكل مرضي فإن المهام المقصودة قد لا تكون قابلة للإنجاز، حتى بواسطة نوع من تحسين الصورة.

إدخال الصور ثنائية الأبعاد 2D

إنّ الصورة الثنائية الأبعاد الأساسية هي أحادية اللون (تدرجات الرمادي) التي حُولت إلى أرقام (صورة رقمية). يمكن وصف الصورة ثنائية الأبعاد كعلاقة رياضية لشدة الإضاءة $f(x, y)$ ، حيث x و y إحداثيات نقطة وقيمة f لأي نقطة (x, y) هي نسبة السطوع (الإضاءة) أو قيمة اللون الرمادي من الصورة في تلك النقطة. الصور الرقمية لها الخصائص التالية:

- قيم إحداثيات وإضاءة منفصلة.
 - كثافة مقاسر عبر إحداثيات شبكة x و y .
 - تم التعبير عن الكثافة بقيم من 8 خانات (256 قيمة).
- للأغراض الحسابية، نحن قد نُفكر بالصورة الرقمية كمصفوفة ثنائية الأبعاد حيث x و y يصنف نقطة من الصورة. كل عنصر في المصفوفة يُدعى نقطة شاشة (عنصر صورة). شاهد الرسم 25 و 26.



الرسم 25. صورة بتدرجات الرمادي وتم تحديد منطقة (المربع الأبيض)

99	71	61	51	49	40	35	53	86	99
93	74	53	56	48	46	48	72	85	102
101	69	57	53	54	52	64	82	88	101
107	82	64	63	59	60	81	90	93	100
114	93	76	69	72	85	94	99	95	99
117	108	94	92	97	101	100	108	105	99
116	114	109	106	105	108	108	102	107	110
115	113	109	114	111	111	113	108	111	115
110	113	111	109	106	108	110	115	120	122
103	107	106	108	109	114	120	124	124	132

الرسم 26: قيم نقاط الصورة في المنطقة المحددة

أجهزة وادوات إدخال الصور ثنائية الأبعاد 2D
آلة تصوير تلفزيونية (كاميرا الفيديو)
الإختيار الأول لإدخال الصور ثنائية الأبعاد هو كاميرا الفيديو -- الفاتج (المخرجات) يكون إشارة فيديو:

- التوجيه نحو الصورة الهدف.
- مسح الهدف خطأً بخط أفقياً للشعاع الإلكتروني
- الإشارة الكهربائية تنتج بمزور الشعاع فوق الهدف.
- الإشارة تمثل نسباً كثافة الضوء في كل نقطة.
- الإشارة الجاهزة للإستعمال تمثل إشارة الفيديو.

هذا الشكل من الأداة له عدة أضرار.

محدودية الوضوح

-- عدد محدود من خطوط المسح (حول 625) وإطر (30 او 60 إطار بالثانية)
التشويه--

- تأخر غير مرغوب به بين الإطار والإطار الآخر.
- خرج فيديو لاخطي فيما يتعلق بكثافة خفيفة.
- الهدف غير مستوي (مسطح) احيانا.

الماسح الضوئي

إلى حد بعيد تعتبر أداة التصوير الثنائية الأبعاد الأكثر شهرة.

- يشتمل على مصفوفة خلايا من العناصر الحساسة للضوء.
- تنتج كل خلية تيار كهربائي معتمد على الضوء الساقط عليه.
- الفاتج "الخرج" إشارة فيديو
- أقل تشويه هندسي
- الفاتج إشارة فيديو أكثر خطية.

مخازن الإطر

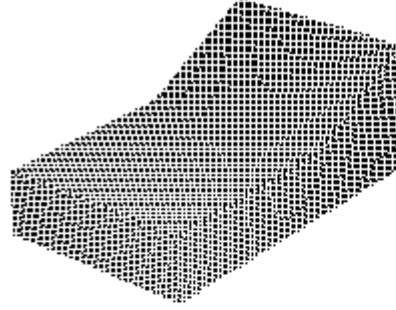
إشارة الفيديو يجب أن تُحوّل إلى أرقام. الأداة المعروفة بمخزن الإطر ومنتزع الإطر يؤدي هذه المهمة حسب التالي:

- يُحوّل إشارة الفيديو المدخلة إلى أرقام
- يقسم الإشارة كعينات إلى نقاط الشاشة المنفصلة في فترات ملائمة -- خط بخط.
- يقسم الإشارة إلى قيم رقمي من 8 خانات.
- يُخزّن عينات الاطر في ذاكرته الخاصة.
- حوّل إطر بسهولة إلى ذاكرة الحاسوب أو ملف.

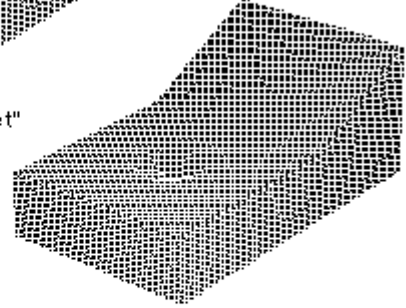
التصوير - الصور ثلاثية الأبعاد 3D

الصور ثلاثية الأبعاد 3D -- خرائط العمق

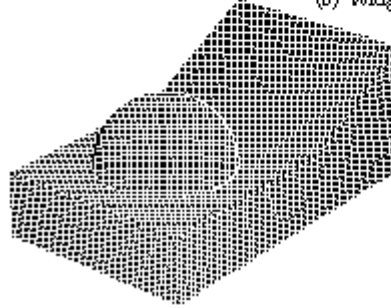
الطريق الأسهل والملائم لتمثيل وتخزين مقاييس العمق من مشهد ما، هو خريطة العمق. خريطة العمق هي مصفوفة ثنائية الأبعاد حيث x و y معلومات إحداثيات المكان المقابل للصقوف والأعمدة كما في الصورة العادية، وقراءات العمق المطابقة (z قيم) مخزونة في عناصر المصفوفة (نقاط الشاشة). خريطة العمق هي مثل الصورة بتدرجات الرمادي، ماعدا z معلومات (حقيقية - 32 بايتات) تستبدل معلومات شدة الإضاءة.



(a) A test object "Widget"



(b) Widget with cylindrical hole



(c) Widget with sphere

الرسم. 27 خريطة عمق إصطناعية



(a) Widget with small coin



(b) Another test object

الرسم. 28 خريطة عمق حقيقية

لماذا نستخدم بيانات ثلاثية الأبعاد؟

الصور الثلاثية الأبعاد 3D لها العديد من الفوائد بالمقارنة مع نظيرتها من ثنائية الأبعاد: الهندسة الواضحة --

- صور 2D تعطي معلومات محدودة فقط للشكل الطبيعي ومقياس الجسم في المشهد.
- صور 3D يعبر عن الهندسة من ناحية الإحداثيات الثلاثية الأبعاد، مما يسمح بتخيل الحجم ومثال على ذلك: - حجم (وشكل) الجسم في مشهد ما يُمكن أن يُحسب ببساطة من الإحداثيات الثلاثية الأبعاد. التقدم التقني الأخير (ومثال على ذلك: - في آلات التصوير الرقمية، آلات المسح الضوئية، أجهزة تعيين المدى الليزرية) سمح بالحصول على معلومات موثوقة ودقيقة لبيانات العمق الثلاثية الأبعاد. لذلك العديد من أنظمة اكتساب البيانات الثلاثية الأبعاد طُوِّرت.

طرق الإكتساب (الإدخال)

أنظمة المدى الليزرية

تعمل هذه الأنظمة على مبدأ انعكاس ضوء الليزر على سطح الجسم، ليقوم حساس ضوئي بإستلام هذا الضوء المنعكس، ثم يقيس الوقت (أو المدة) بين الإرسال والإستقبال لكي يحسب العمق. أغلب المساحات الليزرية لها الخواص التالية:

- تعمل على مسافات بعيدة (15 متر)
- ولذلك قرار عمقهم ناقص لمهام الرؤية المفصلة.
- تم تطوير أنظمة المدى الأقصر لكن ما زال عندها قرار عمق ناقص (1 سنتيمتر في أحسن الأحوال) لأكثر أغراض الرؤية الصناعية العملية.

طرق الضوء المنظّمة

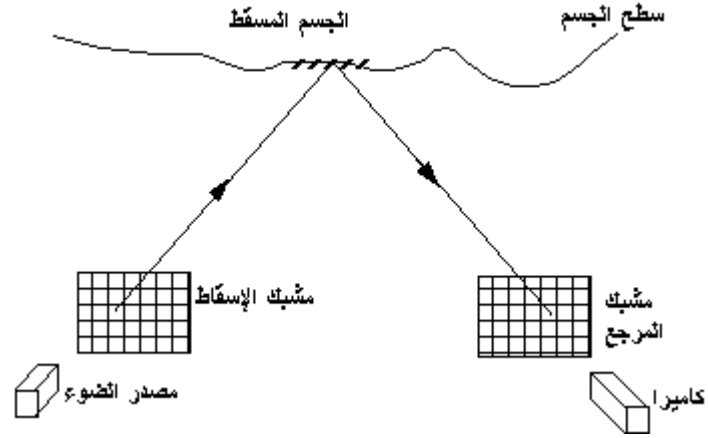
الفكرة الأساسية:

- إسقاط نماذج الضوء (شبكات، أشربة، أنماط إهليلجية الخ.) على الجسم.
- استنتاج الأشكال السطحية من تشويهاات الأنماط التي تُنتج على سطح الجسم.
- العديد من الطرق طُوِّرت ليستعمل هذه النظرية.
- الفائدة الرئيسية - السهولة في الإستعمال.

- الأنماط تُصبح أكثر تناثراً بالمسافة.
- حساسات المدى القريب (4 سنتيمتر) تَجْ عطي قرارِ العمقِ الجيدِ (حول 0.05 مليمتر) لكن لها حقل ضيق جداً للرؤية ومدى قريب من العملية.

طرق Moire الهامشية

إنَّ جوهرَ الطريقةِ بأنَّ يسقط الضوء من خلال مشبك (شبكة) على الجسم المراد تصويره وتتشكل الصورة سطح مشبك آخر مرتبط بالكاميرا (الشبكة المرجعية). تتداخل الصورة مع الشبكة المرجعية لتشكل أنماط محيط مواريه الهامشي الذي يظهر كقطع مظلمة وقطع مضيئة من الأشرطة، كما هو موضح بالرسم. 29. تحليل الأنماط يعطي أوصاف دقيقة من التغيرات بالعمق والشكل.



الرسم 29. نظام إسقاط مواريه

- ملاحظة: تظهر حالات الغموض في استجاب الأنماط الهامشية.
- ليس من الممكن التفرير فيما اذا كانت محيطات الاجسام المجاورة أعلى أو أوطأ للبعق.
 - الحل بتحرك إحدى المشابك وأخذ عدة صور مواريه.
 - المشبك المرجعي يمكن أيضاً أن يُحذف وتأثيره يمكن أن يعدل للبرامج.
- طرق هوامش مواريه قادرة على إنتاج بيانات العمق الدقيقة جداً (درجة الوضوح حوالي 10 ميكرونات) لكن هذه الطرق لها بعض العوائق:
- طرق غالبية بشكل حسابي نسبياً.
 - السطوح في الزوايا الكبيرة غير قابلة للقياس أحياناً -- كثافة هامشية تُصبح كثيفة جداً.

طرق الشكل من الظل

- طرق مستندة على الشكل من الظل استخدام تقنيات مقاييس الضوء الجسم لإنتاج مقاييس العمق. استخدام آلة تصوير واحدة، يمكن اخذ صورتين أو أكثر لجسم في نفس الموقع لكن بشروط إضاءة مختلفة. بدراسة التغيرات في السطوح على السطح واستخدام الهوائي في توجيه السطوح، فإن يمكن حساب معلومات العمق بدقة. الطرق المستندة على هذه التقنيات لا تناسب لجمع بيانات العمق الثلاثي الأبعاد العام:
- طرق معتمدة بشكل حساس على الإضاءة وخصائص الانعكاس على سطوح الأجسام الموجودة في المشهد.
 - تعمل الطرق فقط بشكل حسناً على الأجسام بالقوام السطحي الموحد.
 - من الصعب إستنتاج عمق مُطلق، والتوجيه السطحي فقط يُستنتج بسهولة.
 - طرق تستعمل في الغالب عندما زُغِبْ بِنْتَزاع معلومات الشكل السطحية.

الطرق التجسيمية السلبية

تجسيم الصور كتقنية لقياس المدى بالإسقاط الثلاثي للمواقع المحددة في مشهد صور بالتي تصوير . إن المشكلة الحسابية الأساسية للصور المجسمة أن نجد تطابق النقاط المختلفة في صورتين. هذا يتطلب:

- إستخلاص موثوق لبعض الميزات (مثل الحافات أو النقاط) من كلتا الصور
 - مجارة الهيزات المطابقة بين الصورتين.
 - كلتا هذه المهام غير بديهية وبشكل حسابي معقد.
 - مجسم سلبي قد لا يحتاج خرائط العمق ضمن الوقت المعقول.
 - بيانات العمق المنتجة متناثرة نموذجياً حيث ميزات المستوى العالي، مثل الحافات، مستعملة بدلاً من النقاط.
- ملاحظات:

- مشاكل في إيجاد وتحدثي مكان الميزات بدقة في كل صورة يمكن أن تكون صعبة.
- لتفادي الأخطاء تحتاج لعناية كبيرة.
- مقاييس عمق دقيق إلى بضعة مليمترات.
- مثال على نظام الرؤية الستيريو السلبي TINA طور في جامعة شافيلد.

الطرق التجسيمية النشيطة

مشاكل التقنيات التجسيمية السلبية قد تتغلب عليها بالتالي:

- إضاءة المشهد بمصدر قوي من الضوء (على شكل نقطة أو خط الضوء) الذي يمكن أن يلاحظ بكلتا آلات التصوير.
- نقاط المطابقة المعروفة حددت في كل صورة.
- خرائط العمق يمكن أن تنتج بمسح مصدر الضوء للمشاهد الكامل.
- استخدام مصدر ضوء ليزري يعتبر نموذجياً.
- مجسم فعال يمكن فقط أن يطبق في البيئات المسيطر عليها (المتحكم بها) -- تطبيقات صناعية.

النموذج الهندسي لرؤية الحاسوب

لكي يمثل النموذج الهندسي لرؤية الحاسوب نحن نحتاج لبعض النماذج من أجسامنا. بالنظر إلى أننا نترك معلومات السطح بالعملية البصرية وهذه معتمدة بشدة على هندسة الجسم فإننا نعتبر سمات عن كيفية شكل مثل هذه البيانات. نلاحظ أنه لبعض العمليات البصرية نحن قد نحتاج لإعتبار خصائص الانعكاس للسطح أيضاً. سنلقي نظرة على بعض مخططات التمثيل الرئيسية التي لم تُحدد بالضرورة للرؤية أو مواضيع أنظمة الذكاء الاصطناعي. بعض أنظمة الذكاء الاصطناعي المتعلقة بمخططات التمثيل طور ت. على سبيل المثال الشبكات اللفظية والتي تم مناقشتها سابقاً، وسنناقش تطبيقه في الرؤية لاحقاً (انظر تمييز الأجسام المبني على النموذج)

- نحتاج لبعض أوصاف الجسم (3D) :

o الهندسة.

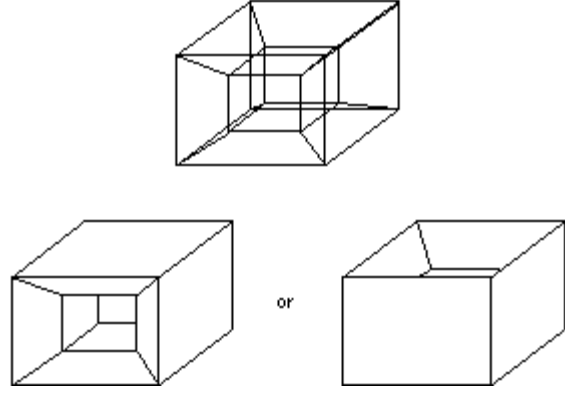
o الهندسة الطبوغرافية.

o ميزات خاصة (ومثال على ذلك: - احتمال الفحص، اللون، القوام)

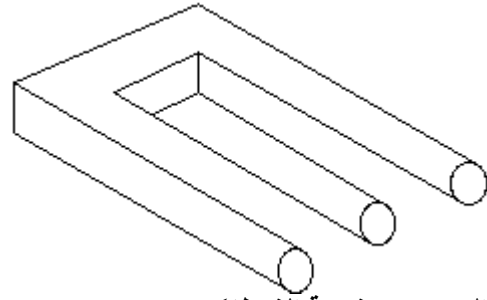
- ميزات مقارنة (دلائل التمييز) إنتزعت من المشهد لتشكيل الوصف للتمييز.
- علاقة بالتصميم بمساعدة الحاسوب (CAD) والرسم الآلي.

نماذج الاطر السلوكية

- تخزين الحافات والقيم في قائمة.
- تمثيل Polyhedral.
- مطابقة مبنية على الجواف.
- مشاكل مع الغموض (الرسم. 30) وتمثيلات مستحيلة (الرسم. 31).
- مشاكل بسبب انه لا مسألة صلبة من الجسم مثلث.
- النماذج الصلبة أفضل.



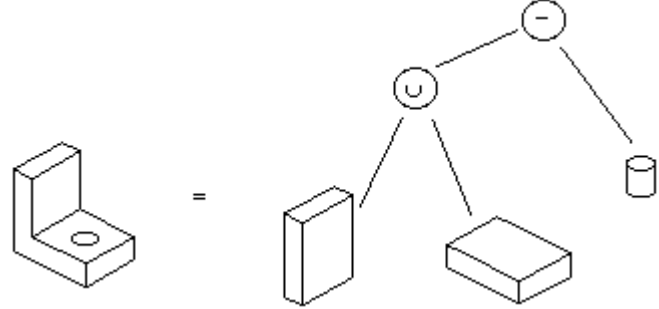
الرسم. 30 إطار سلكي غامض



الرسم. 31 شوكة الشيطان

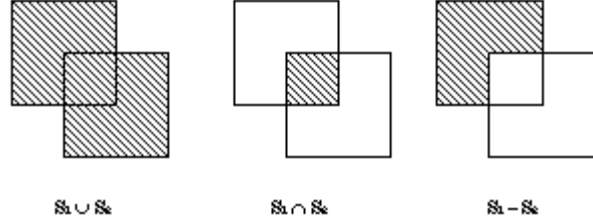
نموذج المجموعة النظري

- أيضاً معروف على نحو واسع بالهندسة الفراغية الحسابية (CSG).
- النموذج متكوّن من الأشكال البدائية، مثل الصناديق المستطيلة والهجالات والإسطوانات والمخاريط... الخ.
- جُمع مع مشغلي المجموعة
- مثل نموذج كتركيب الشجرة (الرسم. 32).



الرسم. 32 نموذج المجموعة النظري

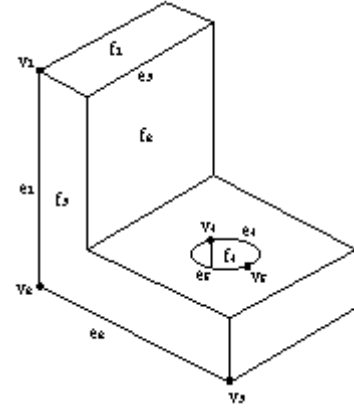
- أسُئلت مجموعة المشغلين المشابهة لتلك التي في الجبر المنطق.
 - المشغلون المستعملون:
- 0 اتحاد -- منطقي، أو
 0 تقاطع -- منطقي، و
 0 إختلاف -- منطقي ليس،



الرسم. 33 مجموعة المشغلين

تمثيل الحدّ

- النماذج بتمثيل الحد لها تمثيل أكثر وضوحاً من نموذج المجموعة النظري.
- الجسم مُمَثَّلُ بتواكيب بياناتٍ معقدة والتي تغطي معلوماتاً حول كلّ من وجوه وحافات وقمم هذا الجسم وكيفية ارتباطهم ببعضهم البعض.
- يبدو انه تمثيل أكثر طبيعي للرؤية حيث ان المعلومات عن السطوح متوفرة بسهولة.
- وصف الجسم يُمكن أن يُقسم إلى جزئين:
الهندسة الطبوغرافية (اللاكمية)
-- يُسجّل ربط الوجوه والحافات والقمم بواسطة المؤشرات في تركيب البيانات.
الهندسة
-- يَصِفُ الشكل المضبوط وموقع كلّ من الحافات والوجوه والقمم.
- قياس القمّة هو فقط موقعها في الفضاء كما هو مُعطى بإحداثياتها (y,z,x).
- الحافات قد تُكون خطوط مستقيمة، أو أقواس دائرية، الخ.
- الوجه مُمَثَّلُ ببعض الوصف لسطحه (جبري أو يستعمل شكلاً مترياً).



الرسم. 34 وجه وحافات وقيم

الخصائص النموذجية المرغوبة للرؤية

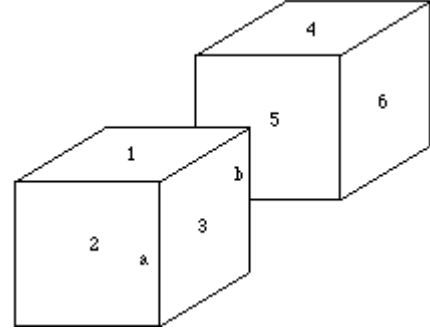
- تطلّبت المهام المثالية عند استعمال نموذج صلب لأغراض الرؤية، هذه المهام تتضمّن:
- تزاوج مباشر بين الميزات النموذجية و ميزات البيانات،
 - تقدير مباشر لموقع وتوجيه الجسم، و
 - تنبؤ ظهور الجسم من أيّ موقع.

تعريف (وصف) الخط

تعتبر قطع الخطوط المستقيمة والمنحنية البنية الأساسية لإغلب الصور. تعطي النسبة الرياضية، بين النقاط المحددة على حدود (محيط) الجسم الموجود في الصورة، التعريف (الوصف) الرمزي للصورة. تعريف الخطّ قابل للتطبيق

عندما تنتزع الخطوطُ الثنائية الأبعادُ من صورة ثنائية الأبعاد. ميدائيًا، نحن سَنُحدِّدُ لأنفسنا الأجسام الهندسية النظامية (المكعب) في المشهد، وبمعنى آخر: . الجسم الذي له وجوهٌ مستويةٌ محاطةٌ بالحافاتِ المستقيمة. تحت هذه الفرضية يمكن تَرْجَمَةُ الحافاتِ:

- لتَحديدِ أي الحافاتِ ربطت بأي وجوه، و
 - لإِسْتِنتاجِ بعضِ العلاقاتِ بين الوجوه.
- لنَعتبرُ رسمَ الخطِّ الظاهر في الوسم. 35.



الوسم. 35 رسم خط من مكعبين

من الممكن أن نرى التالي :

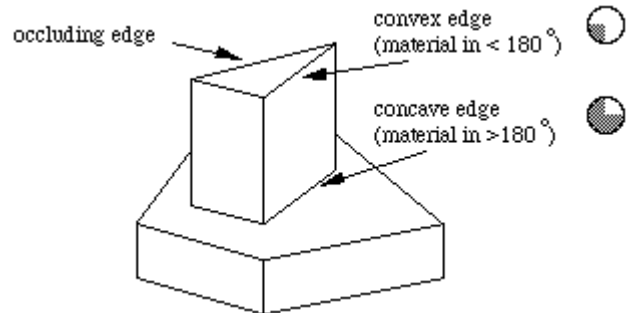
- رَبطت الحافاتِ الوجوه المرقمة، وهي 1، 2 و 3 من المكعب الاول،
- بينما الوجوه 4، 5 و 6 تعود للمربع الآخر (على إفتراض أن المكعبان لم يُلتصقا بخيط دقيق لا يُمكن أن يُرى).

ملاحظة مهمة :

- كل حافة ثلاثية الأبعاد تُرتبطُ بوجهين بالضبط -- واحد على كل جانب منها،
- أحياناً كل من هذه الوجوه يُمكن أن يُرى من وجهة النظر المُعطاة (كما في حالة الخط في الرسم. 36). لو ان أحد الوجوه فقط مرئي – فإن هذه الحافة تُدعى حافة إغلاق. على الجانب الآخر لحافة الإغلاق، الخلفية أو الوجه الغير ملتصق هو مرئي. إن قاعدة طريقة تعريف الخط هي أن نُحاول تصنيف كل حافة في الصورة كما أن تكون إحدى:

- حافة مقعرة (concave)،
- حافة محدبة (convex)،
- حافة إغلاق (occluding).

امثلة على حافات كل نوع موضحة في الرسم. 42.



الوسم. 36 أنواع الحافلت في رسم الخط

حيثما تَجتمعُ الحافاتِ فقط بعضُ التفسيراتِ المحتملةِ لخطوطِ الإجتماع ثابتة بشكل متبادل. هذا يُخفِّضُ عددَ الإمكانياتِ لتصنيفِ كل حافة. إذا الخط يُقابل حافة الإغلاق:

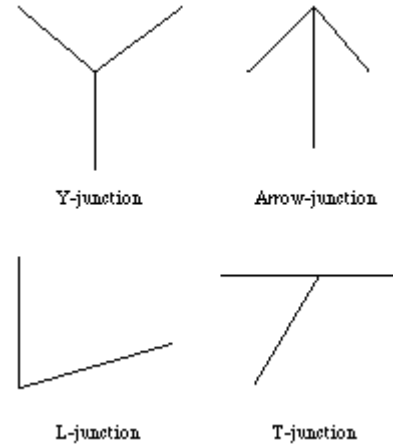
- مُحدّد بسهم. يَعطِي توجيهُ السهم جانِبَ الحافةِ الذي فيه الوجه المرئي، إستعمال التقليد الذي فيه الوجه المرئي يَسْتَنَدُ على الجهة اليمنى مِنَ الحافةِ، نَظَرًا على طول إتّجاهِ السهم.
- الخطوط التي تُقابِلُ الحافاتِ المَحْدَبَةِ محددة بعلامة '+' !.
- الخطوط التي تُقابِلُ الحافاتِ المَقَعَرَةِ محددة بعلامة '-' !.
- كُلُّ حافة في الصورة لها إحد التعاريف الأربعة المحتملة.

الفرضيات

- نحن سَنَعْمَلُ بَعْضَ التَّبْسيطِ للفرضياتِ والذي يَتَعَلَّقُ بِمَنْظَرِ الجسمِ وتَعْقِيدِهِ.
- الجسم في الموقع العامّ – لا يوجد جسمان يَتَراصفانِ عَرَضِيًّا في الصورة بسبب موقع رؤية خاصّ.
 - الحافات الخارجية في الصورة -- صورتها الظلية -- تَعُودُ إلى الوجوه التي تُغْلَقُ الخلفية.
 - الأجسام مِنَ التعقيدِ المحدودِ، وبشكل خاص تلك الحافاتِ الوحيدةِ الثلاثِ تَجْتَمِعُ أبدأً في قَمَّةٍ.

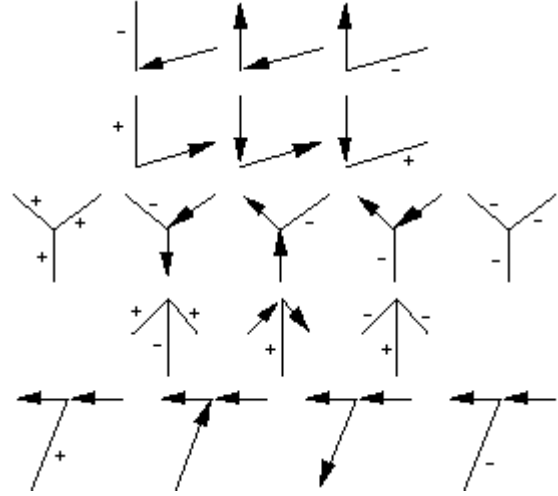
أنواع الملتقى

من المحتمل الآن قَوْلُ ان كُلَّ ملتقياتِ الخطِّ سَنَعُودُ إلى إحدى الأنواع الأربعة، كما هو مَعْرُوضٌ في الرسم. 37.



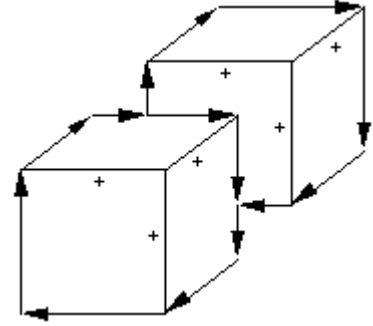
الرسم. 37 أنواع الملتقى في رسم الخط

- في الملتقى من نوع L فقط حافتان تَجْتَمِعَانِ،
 - يَجْتَمِعُ ثلاث حافات في أنواع الملتقى الأخرى.
 - الإمتياز بين ملتقى السهم و ملتقى Y سواء الحافات الثلاث تُقابِلُ زاوية كلّية أقل أو أكثر مِنْ 180° .
 - حالة التّفْسيم بين ملتقى السهم وملتقى Y يُمَيِّزُ بِشكْلٍ مُحدّد ك مفرق T.
- إِعتبار ما يمكن هذه الملتقيات أَنْ تُمثِّلَ من وجهة نظر نموذج الاجسام النظامية يُؤدّي الى استنتاج بأن فقط التعريف المُتأكّد للحافات في الملتقى يُمكن أَنْ يَظْهَرَ في صورٍ مشهَد حقيقي. هذا مُوضح في الرسم. 38.



الرسم. 38 تعريف جانز في الملتقى

هذ الأقل من العدد الكلي لكل التعريف المحتمل. البعض من التعريف الجانز يُظهر الحدّث في المشهد كما في الرسم. 39.



الرسم. 39 الخطوط المعرفة في الصورة

إستنتاج فوري واحد يُمكن أن يُجعل م لتقى T يحدّث فقط عندما جسم ما يُغلق جسماً آخرًا، أو ربما جسم يُغلق جزء من نفسه.

تعريف (وصف) الصورة

- يمكن اعتبار الخطوط المستقيمة والمنحنية كعناصر بسيطة لاجسام اعقد مثل: المثلثات، الدوائر.... الخ. هذه البنى والأشكال ممكن معالجتها وتحليلها وتعريفها بمساعدة خصائص الشكل: القياسية، الطبوغرافية والهندسية. لقرض أنه عندنا صورة خط ونريد تعريفه آلياً. في كلّ قمة
- التعريف يجب أن يكون من التعاريف الجانزة - واحد من العدد الصغير النسبي من الإمكانيات.
- كلّ قطعة خط لها نهايتان، وهي يجب أن يكون عندها نفس التعريف في كلّ نهاية، على الأقل في حالة نماذج الأجسام المنتظمة.
- نبدأ بأخذ الخطوط الخارجية في الصورة ونعتبرهم كإغلاق الخلفية. نحن قد ننظّم طريقتنا الآن كبحت شجرة.
- نختار الملتقى، نعرفه على نحو صحيح،
- نتحرك على طول حافته إلى الملتقيات المجاورة، نعرفهم بطريقة ثابتة.
- إذا هذا ليس ممكناً، فإننا يجب أن نتراجع ونحاول بديلاً آخرًا لإختيار سابق.

- نستمر حتى تعريفنا الرسم بنجاح، أو نوقف البحث في حال انه ليس ممكنا.
- في الحالة الأخيرة، صورة الخط لا تمثل جسم طبيعي صحيح.
- نلاحظ انه قد يوجد أكثر من تفسير صحيح واحد، لذا نحن نريد جعل بحث الشجرة شامل لإيجاد تعريف جائز آخر، بدلاً من التوقف بعد إيجاد تعريف ثابت واحد للصورة الكاملة.
- تحقيق التقييد: هو طريقة تطبيق مشتركة لتعريف الخط.

تعريف الشكل

تقنيات تعريف الشكل يُمكن أن تستخدم في العديد من مجالات رؤية الحاسوب. طرق الشكل استخدمت في العديد من المجالات الأخرى من الحساب عموماً، خصوصاً إلى حل المعادلات اللاخطية الآنية. نحن سنصّف أولاً المبادئ الأساسية وراء طرق تعريف الشكل وبعد ذلك نناقش تطبيقاته المختلفة.

إن العناصر الأساسية لطريقة تعريف الشكل هي مجموعة الميزات التي تعود إلى جسم و مجموعة العلامات. ضمن سياق الرؤية، هذه الميزات عادة هي نقاط وحافات و سطوح. عادة، مخططات التعريف المستعملة فرضية لكل ميزة أو أوزان أو احتمالات يُخصّصان إلى كل علامة في المجموعة التي تُعطي تخمين الإمكانية بأن العلامة المعيّنة صحيحة لتلك الميزة.

النظريات الاحتمالية تُستعمل لتزويد (أو تُقلّل) الاحتمالات بالتعديل التكراري، يأخذ في الحسبان ان الاحتمالات إرتبطت بالميزات المجاورة. إستراتيجيات الشكل لا تضمن تقارب بالضرورة، وهكذا، نحن قد لا نصِل إلى حل تعريف نهائي مع علامة فريدة سيكون عندها احتمال واحد لكل ميزة. دعنا نعتبر نظرية التعريف الآن في التفصيل الأكثر. دعنا نفترض:

- $\{a_1, \dots, a_n\}$ مجموعة من n ميزات جسم للتعريف.
- $\{l_1, \dots, l_m\}$ مجموعة من m علامات محتملة للميزات.
- $P_i(l_k)$ دغ يكون الاحتمال بأن العلامة l_k هي العلامة الصحيحة لميزة الجسم.
- $0 \leq P_i(l_k) \leq 1$ حيث $P_i(l_k) = 0$ يُشير ضمناً إلى أن العلامة مستحيلة للميزة
- $P_i(l_k) = 1$ وتُشير ضمناً إلى أن هذا التعريف مُتأكد
- مجموعة العلامات مشتركة وشاملة. هكذا نحن قد نكتب لكل i :

$$\sum_L P_i(l_k) = 1.$$

وهكذا كل ميزة تُوصف بشكل صحيح بالضبط بعلامة واحدة من مجموعة العلامات. نبدأ عملية التعريف بمهمة أولية، وربما إعتباطية، وهي مهمة الاحتمالات لكل علامة لكل ميزة. الخوارزمية الأساسية تحول هذه الاحتمالات إلى مجموعة جديدة طبقاً لبعض جد أول الشكل. هذه العملية متكررة حتى طريقة التعريف تتلاقى أو تستقر. هذا يتم عندما لا يحدث تغيير أو انه تغيير قليل جدا بين المجموعات المتعاقبة من قيم الاحتمال. يُعتبر مشغل توافق الاحتمالات العلامة للقيود في خوارزمية التعريف.

إن التوافق $C_{ij}(l_k, l_i)$ هو إرتباط بين العلامات المعرّفات للاحتمال الشرطي ان الميزة a_i لها علامة l_k أعطت تلك الميزة a_j لها علامة l_i ، وبمعنى آخر: $C_{ij}(l_k, l_i) = P(l_k | l_i)$

وهكذا، تم تحديث احتمالات العلامات بإعتبار احتمالات العلامات للميزات المجاورة. دعنا نفترض بأننا غيرنا كل الاحتمالات إلى بعض الخطوات، S ، ونحن الان نُبحث عن احتمال مُجدد للخطوة التالية $1+S$.

نحن يُمكن أن نُخمن تغيير $P_i(l_k)$ بسرية بـ:

$$\delta P_i(l_k) = \sum_{j \in N} w_{ij} \left[\sum_{l \in L} C_{ij}(l_k, l_i) P_j(l_i) \right]$$

حيث أن N هي مجموعة الميزات المجاورة لـ Q_i و w_{ij} العامل الذي يزن تعريف هؤلاء الجيران، عرّف بطريقة

$$\sum_{j \in N} w_{ij} = 1.$$

بحيث أن :

الإحتمال الجديد للعلامة $P_i(l_k)$ في الجيل $S + 1$ يُمكن أن يُحسب من القيم من الجيل S بالاستعمال

$$P_i(l_k) = \frac{P_i(l_k) [1 + \delta P_i(l_k)]}{\sum_l P_i(l_i) [1 + \delta P_i(l_i)]}$$

طرق الشكل الإحصائية

طرق الشكل استخدمت خصوصاً لتحقيق الأمثلية في المشاكل. مثل هذا النوع من المشاكل م تداول بكثرة في رؤية الحاسوب ولذا طرق الشكل طُبِّقَتْ بتنوع عريض من طرق إلى رؤية الحاسوب. طرق الشكل استخدمت في:

إرتباط الحافة

-- إحتتمالات نقاط الحافة تقع على الحافات المعيّنة المُحدَّدة باعتبار نقاط الحافة المُجاورة. العلامات المختلفة مستعملة لكل حافة، و جدول الشكل يُستعمل لإيجاد العلامة الملائمة لكل نقطة حافة.

تقنيات تعريف الخط

يُمكن أن يُبدي مشكلة الشكل.

- خطوط العلامة كإتتماء إلى صنف أكيد من الحافة (إغلاق، مقعر أو محدب).
- إحتتمالات يُمكن أن تُخصَّص إلى كل نوع تعريف بانصاف بسهولة.
- فقط بعض مجموعات تعريف الحافة متوافقة بشكل متبادل في ملتقيات الخط.
- هذه القيود يُمكن أن تُبدو كالتقييد عند تخمين الإحتتمالات الشرطية للعلامات المعيّنة.

التقسيم

يُمكن أن يُفسر بطريقتين مختلفتين قليلاً:

- عملية تجميع نقاط الشاشة في مجموعات فرعية تحتوي كل منها على النقاط المتشابهة ضمن شروط معينة. $region_1, \dots, region_n$ إن المناطق تُعرف ببساطة كـ.
- عملية تعريف مناطق (المجموعات الفرعية) الصورة حسب انتمائها الى كيانات طبيعية معرفة (مميزة) مثل: السماء والعشب والأشجار والسيارة والطريق... الخ.

تمييز (التعرف على) الأجسام

مرحلة التمييز هي الهدف النهائي للكثير من أنظمة الرؤية -- يحتاجه إلى:

- فهم الاشياء والاجسام المراد تمييزها.
- للتحرك بسلامة وتجنب الأشياء،
- النقاط ووضع الأجسام المختلفة،
- تفتيش وفحص الأجسام
- لتأدية العديد من المهام الأخرى.

الثوابت

نحن يُمكن أن نستعمل أوصاف بسيطة (قيمة عددية واحدة) لتمثيل الميزات ، وتسمى معايير المطابقة، ومن ثما التعرف على الأجسام:

- يُزوّدون بوسائل بسيطة من المقارنة.
- يُمكن أن يُزوّدوا بمعلومات التوجيه والموقع.

- بسيط جداً لبعض التطبيقات
- لا يعطي وسائل فريدة من المطابقة.
- مثل هذه الأوصاف العددية أو الإحصائية قد تُقسّم إلى صنفين مُتميّزين:
- أوصاف هندسية: المنطقة، الطول، الحافة...الخ.
- أوصاف طوبولوجية: الربط وعدد يولر (Euler)..الخ.
- البعض من المقاييس أعلاه لها معاني واضحة. الأخرى سَتُستَخدم باختصار لاحقاً ضمن سياق الصور الثنائية الأبعاد.
- الإطالة

-- أحياناً تسمّى اللا مركزية أو الشذوذ عن المركز. هذه نسبة الطول الأقصى للخط أو الوتر الذي يغطّي المنطقة إلى الطول الأدنى للوتر. نحن يُمكن أن نُعرّف هذا أيضاً من ناحية اللحظات كما نحن سنرى بعد قليل.

الضغط

-- هذه نسبة مربع الحافة (المحيط) إلى مساحة المنطقة.

عزم القصور الذاتي

-- عزم ij th المنفصلة المركزية m_{ij} لمنطقة مُعرّفة من قبل

$$m_{ij} = \sum (x - \bar{x})^i (y - \bar{y})^j \quad (1)$$

حيث الجمع مأخوذ على كلّ النقاط (x, y) المحتواة ضمن المنطقة و (\bar{x}, \bar{y}) مركز ثقل المنطقة:

$$\bar{x} = \frac{1}{n} \sum_x x \quad \text{and} \quad \bar{y} = \frac{1}{n} \sum_y y.$$

ملاحظة أن n ، العدد الكليّ للنقاط المحتواة في المنطقة، هي مقياس لمنطقته. نحن يُمكن أن نُشكّل سبع عزوم جديدة من العزم المركزي التي تكون ثابتة بالنسبة إلى التغييرات في الموقع والمقياس والتوجيه لجسم ممثلاً بالمنطقة (ثوابت ستناقش لاحقاً بعد قليل)، بالرغم من أن هذه العزوم الجديدة ليست ثابتة تحت الإسقاط المنظوري. العزوم السبع هي كالتالي:

$$\begin{aligned} M_1 &= m_{20} + m_{02} \\ M_2 &= (m_{20} - m_{02})^2 + 4m_{11}^2 \\ M_3 &= (m_{30} - 3m_{12})^2 + (3m_{21} - m_{03})^2 \\ M_4 &= (m_{30} + m_{12})^2 + (m_{21} + m_{03})^2 \\ M_5 &= (m_{30} - 3m_{12})(m_{30} + m_{12}) [(m_{30} + m_{12})^2 - 3(m_{21} + m_{03})^2] \\ &\quad + (3m_{21} - m_{03})(m_{21} + m_{03}) [3(m_{30} + m_{12})^2 - (m_{21} + m_{03})^2] \\ M_6 &= (m_{20} + m_{02}) [(m_{30} + m_{12})^2 - 3(m_{21} + m_{03})^2] \\ &\quad + 4m_{11}(m_{30} + m_{12})(m_{03} + m_{21}) \\ M_7 &= (3m_{21} - m_{03})(m_{12} + m_{30}) [(m_{30} + m_{12})^2 - 3(m_{21} + m_{03})^2] \\ &\quad - (m_{30} - 3m_{12})(m_{12} + m_{03}) [3(m_{30} + m_{12})^2 - (m_{21} + m_{03})^2] \end{aligned} \quad (2)$$

نحن يُمكن أن نُعرّف لا مركزية أيضاً باستخدام العزوم ك

$$\text{eccentricity} = \frac{m_{20} + m_{02} + \sqrt{(m_{20} - m_{02})^2 + 4m_{11}^2}}{m_{20} + m_{02} - \sqrt{(m_{20} - m_{02})^2 + 4m_{11}^2}}. \quad (3)$$

نحن يُمكنُ أَنْ نَجِدَ محاور رئيسية أيضاً مِنْ القصور الذاتي التي تُعرَّفُ نظام إحداثيات طبيعي لمنطقة. دُغ θ نَكونُ مُعطاة بـ

$$\theta = \frac{1}{2} \tan^{-1} \left[\frac{2m_{11}}{m_{20} - m_{02}} \right] \quad (4)$$

نحن سَنَحْصِلُ على قيمتين لـ θ اللتان بعيدتان 90° عن بعضهما البعض. إنَّ زوجَ الخطوط الذي يَعملُ زاويةً θ بالمحور x السيني الذي يَعبُرُ مركزَ ثقل المنطقة يُعرَّفُ زوج المحاور الرئيسية الذي يَصْطَفُ عموماً بمحاذاة ما نسميه بشكل حدسي طول وعرض المنطقة.

الربط

-- عدد الميزات المجاورة التي تُجاوُرُ المنطقة.

عدد يولر Euler

-- لمنطقة واحدة، واحد ناقص عدد الفتحات في تلك المنطقة (= 1 - عدد الفتحات). عدد يولر لمجموعة المناطق المرتبطة يُمكنُ أَنْ يُحْسَبَ كعدد المناطق ناقص عدد الفتحات (عدد يولر = عدد المناطق - عدد الفتحات). أغلب هذه المقاييس الإحصائية قد تُكون مفيدة للمُساعدَة في التحكم بعملية الاستنتاج والتعرف على الصور، حتى وإن كانوا بشكل منفصل غير كافيين لعمل ذلك، بمعنى التفكير والاستنتاج والتمييز. على أية حال، هذه الإجراءات رخيصة جداً للحساب كنواتج عرضية مِنْ إستراتيجيات التفكيك المفصلة، خصوصاً نمو منطقة. ولذلك، عند استعمال هذه الإجراءات بالارتباط مع الميزات المُنتزعة بشكل موثوق الأخرى، هم يُمكنُ أَنْ يُساعدوا لتحسين تقنيات التعرف والتمييز بتضييق مجال البحث للم تطابقات بين الصورة والميزات الجسم المراد التعرف عليه.

الإجراءات الثابتة

عدّة مِنْ المقاييس الإحصائية الذي مرت سابقاً هي إجراءات ثابتة، التي تُقَوَّلُ بأنَّ قيمة الإجراءات لا تَتَغَيَّرُ، على سبيل المثال، موقع المنطقة في الصورة، أو ربما توجيهها أو مقياسها. هذه بشكل واضح خاصية جيدة حيث أننا لا نَسْتَطِيعُ ضمان رؤية جسم مِنْ وجهات النظر المماثلة. وهكذا، بينما مركز ثقل منطقة من الواضح يَتَغَيَّرُ بموقعه، بينما منطقته ثابتة. بينما المنطقة تتفاوت بالمقياس (على سبيل المثال، قرب آلة التصوير إلى الجسم)، التركيز كما عُرِفَ اعلاه ثابت فيما يتعلق بالمقياس بالإضافة إلى الموقع والتوجيه. الإجراءات الثابتة الأخرى:

- أوصاف التقوس السطحي -- تقوس Gaussian والتقوس المتوسط هما ثابتان فيما يتعلق بدوران وتحريك السطح.
- عزم القصور الذاتي $M_1 \dots M_7$
- هو ثابت فيما يتعلق بتغيير المقياس (التكبير و التصغير) والدوران والتحرك.
- الا مركزية ثابتة أيضاً، فيما يتعلق بتغيير المقياس (التكبير و التصغير) والدوران والتحرك.
- تحويلات فوريير (Fourier) ثابتة بالنسبة للدوران. نحن يُمكنُ أَنْ نَسْتَعملَ تحويل فوريير لحساب واصف فوريير للجسم الذي يعتبر ثابت فيما يتعلق بالموقع والتوجيه.

تمييز النمط

طريق واحد للاكتشاف إذا جسم (أو جزء جسم) أَنْ يَبْحَثَ عن النمط البدائي الواقع في المشهد.

مُطابَقة القالب

مُطابَقة القالب بسيطة في الحالات التي فيها الأنماط المَخْزُنة موجودة في الصورة. مُطابَقة القالب أدبَتْ على مستوى نقطة الشاشة وأيضاً على المستوى الأعلى.

مُطابَقة قالب نقطة الشاشة

هنا نريدُ قالب نقطة الشاشة ذات المستوى المنخفض. هناك 4 مبادئ:

القوالب الكلية

-- هنا مطلوب مُطابَقة مضبوطة. إنَّ القالب نفس الحجم كالصورة المُدخلة. ليس هناك ثابت دوران أو تحريك.

القوالب الجزئية

-- هنا القالب بدون خلفية. الم تطابقات المتعددة مَسْمُوحَة. الم تطابقات الجزئية لَرُبَّمَا أيضاً مَسْمُوحَة. يجب اخذ الحيلة في هذه الحالة - القالب F يُمكنُ أَنْ يطابق بسهولة إلى القالب E. متطلبات التزوي قد تُحدُثُ تمثيلات النمط.

قوالب القطعة

-- لإجراء المطابقة هنا يقسم النمط إلى القوالب المكونة. ومثال على ذلك: - النمط A يُمكن أن يهيز بثلاثة قوالب /،
، - . متطلبات الخزين أقل في هذه الطريقة.

القوالب المرنة

-- هذه القوالب يُمكن أن تُعالج الإمتداد ، التوجيه والانحرافات المحتملة الأخرى. النموذج الجيد لجسم معروف مكتسب أولاً ومثّل بشكل عوامل. الأمثلة الأخرى مُثلة وبعد ذلك يتم تعديل العوامل (التعلم بتعديل العوامل). تم استخدام الطرق المبنية على الإرخاء لتقرير المطابقات المحتملة.

مطابقة القوالب العالية المستوي

المشكلة المتعلقة بنقطة الشاشة هي بالرغم من أن هارخيصرة وبسيطة جداً للتطبيق بالنسبة الدوران والتحريك هي مشكلة. أيضاً الصور تعاني من التشويش، تشوهات ممتدة، وملينة بالضوضاء. ومثال على ذلك: - كيف يمكن لطرق المستوى المنخفض معالجة الأحرف المكتوبة بخط اليد؟

تشتغل طرق مطابقة قوالب المستوى العالي على الصورة التي قُسمت نموذجياً إلى مناطق الإهتمام. المناطق يُمكن أن تُوصف من ناحية: "المساحة، متوسط الكثافة، نسبة تغيير الكثافة، القوس". و قارنت أيضاً - "أكبر من، مجاور ل، فوق، مسافة بين".

القوالب لمُصنوفة في العلاقات بين المناطق. استُخدمت هنا قواعد إنتاج و تمثيلات لغوية أخرى. لتنفيذ المطابقة استخدمت طرق إحصائية أيضاً (تقنيات مبنية على الإرخاء).

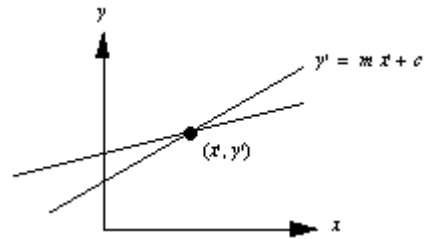
تحويلات Hough

تحويلات Hough يُمكن أن تُعتبر طريقة المطابقة بتعميق القالب. تحويل Hough نموذجياً يُستعمل لإنترزاغ الحواف أو النفوس من الصورة. في هذه الحالة يُمكن إعتباره رابط الحافة حيث أن يُجمع نقاط الحافة سوية ويصنف كمعادلة خط.

على أية حال تحويلات Hough يمكن أن تُستعمل لإنترزاغ الدوائر و الأشكال المعقدة حتى (ربما غير متماثلة). في هذه الحالة هي مثل مُطابق النمط. نلاحظ أن تحويلات Hough ثابتة بالنسبة إلى الدوران والتحريك. المدخلات إلى تحويلات Hough عادة هي صورة التي إكتشفت حوافها - باستخدام أحد كواشف الحافة ، على سبيل المثال. دعنا نفترض بأننا نبحث عن الخطوط المستقيمة في صورة. إذا تأخذ نقطة (x', y') في الصورة، كل الخطوط التي تُعبر تلك النقطة لها الشكل

$$y' = mx' + c \quad (5)$$

لتفاوت قيم m و c . شاهد الرسم 40.

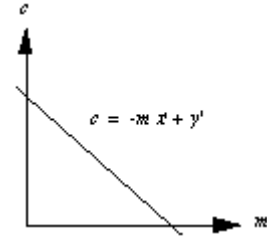


الرسم 40 خط خلال نقطة

على أية حال، هذه المعادلة يُمكن أيضاً أن تُكتب ك

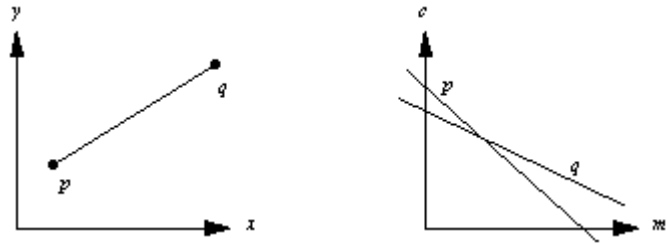
$$c = -x'm + y' \quad (6)$$

حيث نعتبر الآن x و y لهما ثوابت، و m و c لمتغيرات. هذا خط مستقيم على محاور إحداثيات c ضد m كما هو م عروض في الرسم. 41. كل خط مختلف خلال النقطة (x', y') يُقابل إحدى النقاط على الخط في (m, c) المستوي.



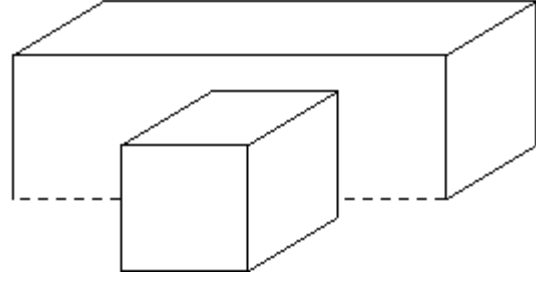
الرسم. 41 خط بإحداثيات m و c

- لنعتبر النقطتين p و q في إحداثيات (x, y) واللذان تقعان على نفس الخط.
- لكل نقطة، كل الخطوط المحتملة التي تمر خلالها ممثلة بخط واحد في إحداثيات (m, c) .
- الخط الوحيد في إحداثيات (x, y) الذي يمر بكل من النقطتين يقع على تقاطع الخطين الممثلين p و q في إحداثيات (m, c) ، كما هو موصوّر في الرسم. 42.



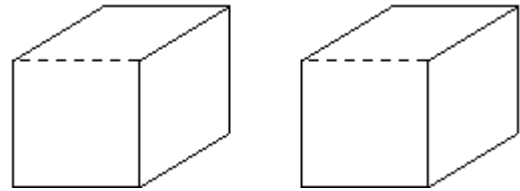
الرسم. 42 نقطة على نفس الخط

- رَأخذ هذه الخطوات،
 - كل النقاط التي تقع على نفس الخط في الاحداثيات (x, y) ممثلة بالخطوط التي تمر خلال نقطة وحيدة في إحداثيات (m, c) .
 - النقطة الوحيدة التي هم جميعاً يَبرون خلالها تُعطي قيم m و c من معادلة الخط $y = mx + c$.
 - لاكتشاف الخطوط المستقيمة في صورة، نحن نعمل التالي:
 - 1. رُئيّت قيمة إحداثيات (m, c) في مصفوفة ثنائي الأبعاد A للخطوات الملائمة من m و c .
 - 2. نعط القيم المبدائية صفر لكل عناصر A (m, c) .
 - 3. لكل نقطة (x', y') تقع على بعض الحواف في الصورة، نُضيف 1 إلى كل عناصر A (m, c) التي تشير الى ان m و c تحقق $y' = mx' + c$.
 - 4. رنحُث عن عناصر A (m, c) الذي له قيمة كبيرة -- كل واحد يتم إيجاده يطابق مع الخط في الصورة الأصلية.
- ميزة مفيدة واحدة من تحويلات Hough هي ان تلك النقاط التي تقع على الخط لا يلزم أن تكون كلياً متاخمة له. على سبيل المثال، كل النقاط الواقعة على الخطين المنقطين في الرسم. 43 سيُكون من متعرف عليهم كوقوع على نفس الخط المستقيم. هذه يمكن أن تكون مفيدة جداً عندما تُحاول إكتشاف الخطوط بينقطاعات قصيرة تحتسب كإشارة ضجيج، أو عندما الأجسام تُغلق جزئياً كما ظهر.



الرسم. 43 نقاط غير متاخمة على نفس الخط

من الناحية الأخرى، هو يُمكن أن يعطي نتائج مَضللة اذا صادف أن الاجسام كانت مُصنَّعة بِالصُدفة، كما هو معروض من قبل الخطان المنقطان في الرسم. 44.

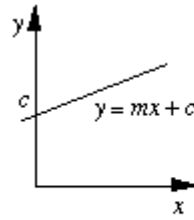


الرسم. 44 جسم مُصنَّف

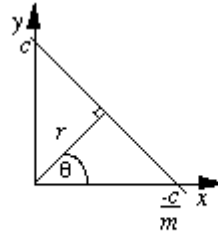
في الحقيقة، هذا يُرى بشكل واضح بأن واحد من عوانق طريقة تحويلات Hough بأنها تُعطي خطاً لانهائياً معبر عنه بزوج النقاط m و c ، بدلاً من قطعة خط محدودة بنقطتنا نهاية واضحتا المعالم. المعادلة $y = mx + c$ شكلت لتمثيل الخط المستقيم الذي يتوقف للخطوط العمودية، عندما تصبح m لانهائياً. لتفادي هذه المشكلة، من الأفضل إستعمال الصياغة البديلة (الرسم 44)،

$$x \cos \theta + y \sin \theta = r, \quad (7)$$

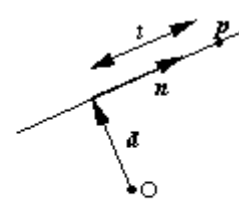
كوسائل وَصَف الخطوط المستقيمة.



(a) $y = mx + c$ equation of line



(b) Alternative form of line equation



(c) Vector representation of a line

الرسم. 44 تمثيل الخطوط

تُلاحظ، على أية حال، ان النقطة في إحداثيات (x, y) نمثل الآن من قبل منحنى في الفضاء (r, θ) بدلاً من خط مستقيم.

ما عدا ذلك، الطريقة بدون تغيير.

كما ذكرنا، تحوُّلات Hough يمكن أن تُستعمل لإكتشاف الأشكال الأخرى في الصورة بالإضافة إلى الخطوط المستقيمة.

على سبيل المثال، إذا نريد إيجاد الدوائر، بالمعادلة

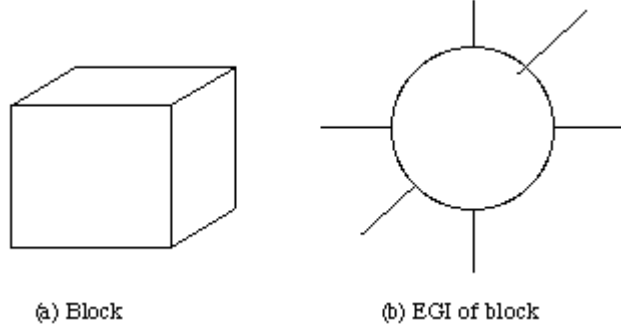
$$(x - a)^2 + (y - b)^2 = r^2, \quad (8)$$

الآن:

- كل نقطة في فضاء (x, y) تقابل سطح في فضاء (a, b, r) (كما نحن يُمكن أن نُغيّر أيّ إثنين a, b, r ، لكن السُّلُوك ثابت).
- الطريقة الأساسية، عدلت لإستعمال المصفوفة الثلاثي الأبعاد (a, b, r) ،
- كل النقاط منها و التي تحقق المعادلة للدائرة مضاعفة.
- عملياً، تأخذ التقنية كميات متزايدة بسرعة من الوقت للأقواس المعقدة الأكثر كعدد المتغيرات (ولذلك عدد أبعاد A). لذا الطريقة جيدة للإستعمال للأقواس البسيطة.

صور Gaussian الممتدة

- معلومات موجه (شعاع) طبيعية سطحية لأي جسم يُمكن أن يمثل كتخطيط وحدة كروية ، نسمي كرة Gaussian.
- التخطيط يُدعى صورة Gaussian للجسم.
- التخطيط: الموجهات لكل نقطة من الجسم موضوعة بحيث :
o الفيول (ذيل الموجه) تقع في مركز كرة Gaussian
o تقع الرؤوس (رأس الموجه) على نقطة على الكرة تُخصّص إلى التوجيه السطحي المعين.
- نحن يُمكن أن نُمدّد هذه العملية لكي نحصل على التالي :
• يخصص وزن إلى كل نقطة على كرة Gaussian ويساوي منطقة السطح التي سيكون عندها الموجه المُعطى
- هذا التخطيط يُدعى صورة Gaussian الممتدة (EGI).
- الأوزان المُتمثلة بالموجهات موازي إلى الأوضاع الطبيعية السطحية، بنظير الطول إلى الوزن.
- مثال على هذا صورة Gaussian الممتدة الواضحة في الرسم 45.

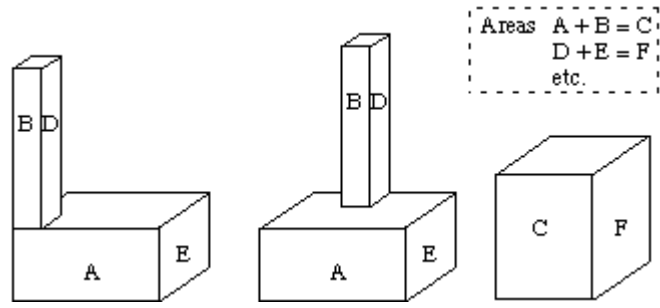


الرسم 45. EGI لكتلة

- إستعمال النماذج الصلبة الثلاثية الأبعاد من الأجسام (ستناقش في القسم التالي)،
- EGIs المطابق لكل نموذج جسم مخزن يُمكن أن يُحسب ويُوفّر في قاعدة البيانات النموذجية
- النموذج مخزن لرسوم بياني إحصائي لموجه طبيعي سطحي.
- إنتزعت الموجهات السطحية بسهولة من الصورة (تركيب مستوي)
- تتم المطابقة (بمقارنة الرسم البياني الإحصائي EGI) وتُحسبان لكل من الهوية وتوجيه الجسم.

العوائق:

- يُعرف EGIs الأجسام المحدبة فقط إستثنائياً.
- عدد لانتهائي من الأجسام غير المحدبة يُمكن أن يملك نفس EGI.



الوسم 46. أمثلة الأجسام بنفس EGI

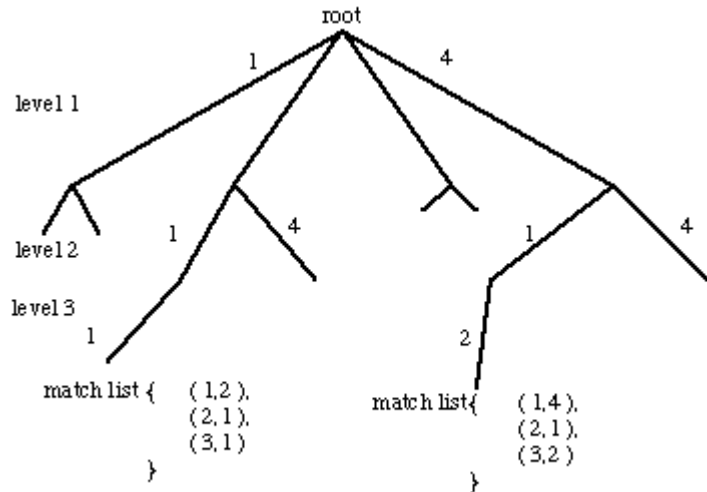
تمييز الاجسام المبني على النموذج

- التمييز هو مشكلة مطابقة بين المشاهد ووصف النموذج.
- المطابقة مشكلة كلاسيكية من مشاكل الذكاء الاصطناعي.
- العديد من طرق الذكاء الاصطناعي المبنية على اساس البحث قدّمت للمحاولة لحل هذه المشكلة.
- يعتبر التمييز معقد ومركّز بشكل حسابي.
- سنناقش ثلاثة طرق واسعة هنا.

طرق بحث الشجرة

النظرة الأساسية:

- عُقد الشجرة تُمثّل المشهد لمطابقة عناصر النموذج الأولية (ومثال على ذلك: - الحافة، السطح...الخ).
- لنفترض ان الشجرة لها عدد m من الفروع في كلّ عقدة التي تُقابل العناصر الاولى للنموذج.
- لنفترض ان مستوى الشجرة يُمثّل العناصر الاولى للنموذج.
- المستوى و موقع العقدة يُحدّدان زوج المطابقة.
- نستعمل إستراتيجية بحث الشجرة (انظر اسفلا للطرق المستعملة) لإيجاد المطابقة.
- اجتياز واحد خلال الشجرة يعطي قائمة المطابقة -- المطابقة المحتملة من المشهد لتشكيل الميزات.
- التمثيل دعا البحث أو شجرة التفسير .



الوسم 47. شجرة التفسير

كَيْفَ تُؤَدِّي الْمُطَابَقَةُ؟

- في كُلِّ مستوى من مستويات الشجرة، أحد الحافات مِنْ المشهد تطابق بِكُلِّ الحافات m المحتملة في النموذج.
 - كُلُّ عقدة لها m أطفال الذين يُمَثِّلُونَ أَخْذَ الم طابوق المقترحة الى ابعد حد مع كُلِّ المطابقات المحتملة لحافة المشهد الحالية.
 - التحويل الذي يُمَثِّلُ الم طابوقَ حتى الآن يُمَكِّنُ أَنْ يُبْقَى.
 - شجرة بحثٍ بلسلوب العمق أولاً والنشذبيَّ برَفْضِ التفسيراتِ التي لا تحقق الم طابوقَ الحالية.
 - فضاء البحث كبيرٌ -- m^n مجموعات محتملة لعدد n من العناصر الاولية للمشهد.
 - الكثير مِنْ الحسابات؟
- نحن يُمَكِّنُ أَنْ نُخَفِّضَ المتاعب الحسابية بِاسْتِخْدَامِ بَعْضِ القيودِ الهندسية المحلية لِتَشْذِيبِ الشجرة أَكْثَر.
- وهي :

- سهولة للحساب والاستخدام.
- تطبيق قبل إختبار التحويل.

للحافاتِ نحن يُمَكِّنُ أَنْ نَسْتَخْدِمَ:

تقييد المسافة

-- طول الحافةِ المُحَسَّنةِ يَجِبُ أَنْ يَكُونَ أَقْلَ أو يساوي طولِ الحافةِ النموذجيةِ قَيِّدَ النَّظَرِ.

تقييد الزاوية

-- الزاوية بين حافتين مُحَسَّنَتَيْنِ متجاورتين يَجِبُ أَنْ تُوافِقَ تلك بين الحافتين المتناظرتين في النموذج المطابق.

تقييد الإتجاه

-- d_{ab} دَعْ يُمَثِّلُ مدى الموجهاتِ مِنْ أَيِّ نقطة على الحافةِ المُحَسَّنةِ a إلى أَيِّ نقطة على الحافةِ المُحَسَّنةِ b . في التفسير الذي أَحَسَّ على التوالي أزواج الحافاتِ a و b مع حافاتِ النموذج i و j ، هذا المدى مِنْ الموجهاتِ يَجِبُ أَنْ يَكُونَ متوافق بـمدى الموجهاتِ المَنتَجَ مِنْ قِبَلِ i و j .

يُمَكِّنُ أَنْ نَسْتَخْدِمَ للسطوح:

- زوايا بين المستويات (السطوح).
- منطقة السطوح.
- ثوابت.
- إجراءات التقوس.

دراسات بعض حالات المطابقة المماثلة

طور العديد مِنْ القيودِ لكي تُنتَجَ موجه مساعد لبحث كفاءة. هنا نَصِفُ بَعْضَ الطرقِ المشهورة التي تُصَوِّرُ تشكيلة واسعة من الموجهات المساعدة.

طريقة **Oshima و Shirai (1979)**

هذه الطريقة كانت واحدة من أوّل الطرق التي طُوِّرتُ للتعامل مع تشكيلة أصناف الأجسام ولكي تَكُونَ قادرة على التعرف على مشهد وَصَفَ بخرائط العمقِ يَحْتَوِي على أجسامَ متعددة في أَيِّ توجيه. الأجسامُ أَعْتَبِرَتْ بَلْبُ يَكُونَ عِنْدَهَا سطوحٌ مستوية، سطوحٌ مُقَوَّسة بيسر أو كلا النوعين. أصناف السطوح المُقَوَّسة بيسر سَمَحَتْ لِنَتَضَمَّنِ مجسمات القطع الناقص، مخاريط ، وإسطوانات ..الخ.

تَعْمَلُ هذه الطريقة بمرحلتين.

مرحلة التعلّم

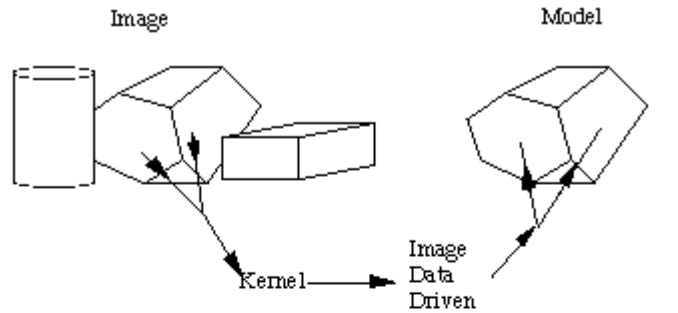
هذه الطريقة لا تَسْتَعْمَلُ أَيَّ تمثيل نموذج صلب مَخْزُونٍ لإدلاء الم طابوق. بدلاً من ذلك تستخدم نماذج مَبْنِيَة مِنْ صور الأجسامِ الَّتِي سَتُعْتَرَفُ عليها. وهكذا ، مرحلة التعلّم تَحْدُثُ في الحالات التي بها الأ جسامُ المعروفة تُعْرَضُ بِشَكْلِ منفصل إلى النظام من وجهات نظر مختلفة. وصف كُلِّ جسمٍ يُعَزَّزُ، من ناحية خصائص السطوح وعلاقاتهم، لَخْلُقِ نموذج الجسم.

إنَّ الوصفَ يكون من إستعمال نفس السطوح المُقَوَّسة والمستوية من الصور الفردية. إنَّ طريقة الإنقسام إستخدمت من قبل **Oshima** و **Shirai** هي طريقة على اساس نمو ال منطقة، بخلاف ذلك يحسب الكثير من خصائص السطوح والعلاقات الداخلية. ثم يتم تضمين هذا في رسم بياني علائقي أو شبكة سيمانطيقية (لفظية). الخصائص السطحية المثالية لكل منطقة تتضمن:

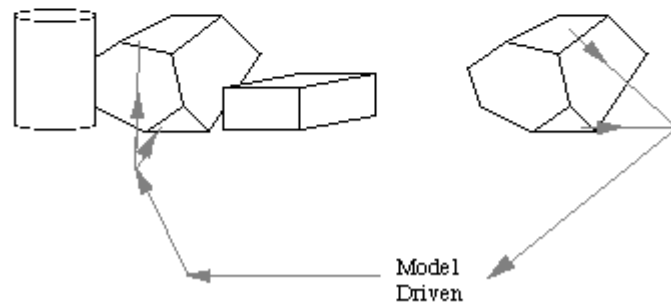
- نوع سطحي (مستوي، إسطوانة الخ.)،
 - عدد السطوح المجاورة،
 - المنطقة،
 - الحافة،
 - الحد الأدنى، الحد الأقصى، أنصاف الأقطار المتوسطة من سطح كروي غير منتظم.. الخ
 - الانحراف المعياري من أنصاف الأقطار من سطح كروي غير منتظم.
- إنَّ العلاقات بين السطوح مُمَيَّزة بالتالي:
- المسافة بين سطح الاشكال الكروية الغير منتظمة ،
 - الزاوية بين أفضل المستويات اللاتقة والمناطق المجاورة
 - نوع التقاطع -- مصنف كمقعر أو محدب، أو خلعي (إذا التقاطع ليس محدب أو مقعر) أو لا يوجد تقاطع.
- فقط مجموعة منفصلة من وجهات النظر تُستعمل لنمذجة الجسم. لوصف كامل لنموذج جسم يراد نجمي، كل المشاهد الفريدة للجسم يجب أن تُقدَّم إلى النظام.

مرحلة المطابقة

تعمل هذه المرحلة بمحاولة مُطابق مجموعة المناطق المرئية الملاحظة في المشهد إلى كل منظر جسم في قاعدة بيانات أوصاف الجسم المتعلمة. ليس كل السطوح موجودة في نموذج المطابقة للمشهد قد تكون موجودة في وصف المشهد المُلاحظ بسبب التفسير السيئ، لذا الإجراء الذي يُحدِّد المطابقات الخاطئة مطلوب. يُجَزَّ هذا بإطلاق البحث الأولي للبيانات المُلاحظة وبعد ذلك، عندما تطابق مجموعة مناسبة من ميزات المشهد بنجاح بالنموذج، بيانات النموذج تُستعمل لتوجيه بقية البحث بشكل كفوء (انظر الرسم. 56)



(a) Image driven first phase of match



(b) Model driven second phase of match

طريقة 3DPO (1983)

الـ 3DPO (توجيه جزء ثلاثي الأبعاد) هو نظام رؤية طَوَّرَ مِن قِبَل Bolles وزملائه للتعرف وتحديد مكان الأجزاء مِنْ خرائط العمق لأغراض توجيه ذراع آلي لِإِلتقاط تشكيلة واسعة من الأجزاء الصناعية المعقَّدة باعتدال. لوصول إلى هذا الهدف تبين بأن أكثر إستراتيجيات الرؤية آنذاك لَمْ تُرَوِّدْ بلعمومية المطلوبة، أَنْ تُكُونْ مستندة على أنواع الجسم العام البسيط مثل الأسطوانات، الكرات... الخ. بدلاً من ذلك تَبَيَّنوا طريقة مختلفة كلياً والتي يُمكنُ أَنْ تتعرفَ على الأجسام بِإِستعمَالِ فقط بضعة ميزات أو ميزات العناقيذ التي تُضيقُ عدد الاجسام إلى حدٍ كبير (مثل دائرة مع نصف قطر ثَبَّت) وبالتالي تقليل حجم فضاء بحثِ الحَلِّ. هذه الأفكار الآن المعروفة بطرقِ بؤرة الميزة المحلية. العملية المماثلة تُشملُ خمس مراحل:

كشف ميزات العنصر الاولي

-- ميزات المستوى المنخفض تُقَسَّمُ وتُرتبُ سوية. تكتشف الحافات في البيانات الثلاثية الأبعاد. تصنف الحافات إلى مقعرة، محدبة، أو حافات خطوة.

تشكيل عنقود الميزات

-- هدف هذه المرحلة أَنْ تُشكَّلَ عناقيذ (العناصر الأولية) الميزات التي يمكن أَنْ تُستعملَ لإفترض النوع وموقع وتوجيه الجسم.

توليد الفرضيات

-- لإيجاد المطابقة يعمل بحث قاعدة بيانات النماذج. الإستراتيجية المستخدمة هنا أَنْ رَجَتِبَرِ العناقيذ المفردة أولاً وإذا هذه ليست كافية، يختبر مجموعات العناقيذ. في هذه المرحلة تستعمل نظرية بؤرة الميزة بالبحث عن جسم من عناقيذ معينة، مثلاً دوائر ذات نصف قطر مُحدد.

إثبات الفرضية

-- الفرضيات المُشكَّلة تُجَرَّبُ بالبحث عن الميزات الإضافية في الصورة المُتَّسقة مع كُلِّ فرضية مُعطاة.

تصفية العوامل

-- في هذه المرحلة النهائية، الفرضية المُحَقَّقة المعطاة، مُستَوجبة ثانيةً لكي تُنتج موقع أكثر دقة للجزء.

طريقة المختصر (ACRONYM) (1979)

طور بروكس نظام رؤية ، *ACRONYM* ، الذي يُعرفُ على الأجسام الثلاثية الأبعاد الواقعة في الصور الثنائية الأبعاد. الأمثلة تتضمن التعرف على الطائرات على مدارج المطار مِنْ الصور الجوية. يُستعملُ مبدأ الأسطوانة المعمَّم لتمثيل كلا من النموذج المخزون والجسم المُنتزع مِنْ الصورة. كما في طريقة Nevatia و Binford (انظر القسم القادم) ، استخدم تركيب الوسم البياني العلائقي لتَحْزِينِ التمثيل. العُقد هي الأسطوانات المُعمَّمة والصلات يُمثِّلان التحولات النسبية (دوران وتحريك) بين أزواج الأسطوانات.

يُستعملُ النظامُ تركيباً رسم بياني آخرين أيضاً، التي هي مَبْنِيَّة مِنْ نماذج الجسم، للمُساعدَةِ في إستراتيجية المطابقة:

مخطط (رسم بياني) التقيد

-- هذا المخطط (الرسم البياني) يُمثِّلُ القيودَ على أصناف الأجسام، لكي يَجْعَلَ الجزء بالالوصاف العامة أوصافاً أكثر تعييناً. أعطى مثال مِنْ قِبَل بروكس الذي يُعتبرُ أصنافَ المحركات الكهربائية. هَذَا يُمكنُ أَنْ يُوصَفَ بنوع المحرك العام الذي يُقسَّمُ إلى الأصناف الأكثر تعييناً مِنْ المحركاتِ مثل محرك مع قاعدة، محرك بالحافات. هذه يُمكنُ أَنْ تُكُونُ مصنفة من ناحية الأصناف الوظيفية (معتمدة على الإستعمال) مثل مضخة ماء التدفئة المركزية أو مضخة الغاز.

القيود الإضافية مَسْمُوحَةٌ لكي تُضافَ إلى المخطط أثناء عملية التعرف.

مخطط التنبؤ

-- الصلات في هذا المخطط نُقْتَلُ العلاقات بين الميزات في الصورة. إِنَّ الصلات معرفة : " يَجِبُ أَنْ تُكُون "، " سِيَكُونُ " أو " مقصور على " طبقاً لإحتمالية حدوث، وبوقت واحد، زوج الميزات المعطاة في جسم وحيد.

حيث ان الصورة حساب ثنائي الأبعاد فقط يَجِبُ أَنْ يَكُونُ مأخوذاً مِنْ إسقاط ميزات الجسم الثلاثي الأبعاد إلى بُعدين. يُستعملُ بروكس وصفان للميزات الثنائية الأبعاد:

الاشرطة

-- تُشتملُ لوصف مساقط شكلا م كون من الإسطوانات المعممة. نلاحظ ان Nevatia و Binford إستعملا شكل ثلاثي الأبعاد من الشريط.

القطوع الناقصة

-- تُشتملُ لوصف مساقط نهايات إسطوانة مُعمّمة. مساقط نهايات إسطوانة دائرية هي قطوع. للمقاطع العرضية المتعددة الأضلاع، القطوع يُمكن أن تُعطى وصف النهايات بتركيب أفضل دائرة أو قطع ناقص خلال القيم وملاحظة مسقط هذا الشكل.

إن عملية المطابقة تؤدي على مرحلتين:

1. نبحث في الصورة عن الم طابقات المحلية لإشتقاق الأشرطة من النموذج المخزون. في مثل هذه الحالات من المطابقات، الأشرطة مُجمّعة في عناقيد.
2. تفحص العناقيد من أجل الإتساق العام بما في كل م طابق يجب أن تحقق كلتا قيود مخطط التنبؤ، والقيود المتراكمة لمخطط التقييد.

هذا العمل وسع من قبل Kuan و Drazowich للسماح لإستعمال الصور الثلاثية الأبعاد. إن المبادئ الأساسية للمختصر ملتزم بها لكن الخصائص السطحية مُدمجة. أيضاً، مقاييس الميزة العددية مستعملة أثناء المطابقة (بينما هناك إستعمال ميزات رمزية فقط في الطريقة الأصلية). إن الخصائص السطحية مُتضمنة في التمثيل المتعدد المستويات للأجسام المستعملة في مرحلة التنبؤ. إن المستويات والميزات مثلاً في كل مستوى:

مستوى الجسم

-- يخرن، على سبيل المثال، العلاقات المكانية بين مكونات الجسم، أبعاد الجسم، خصائص نقطة النظر مثل الحجم الأدنى والأقصى، وعلاقات الإمتصاص بين الميزات.

مستوى الإسطوانة المُعمّمة

-- يخرن المحيط الحجمي للإسطوانة وموقع الإسطوانة والتوجيه، وأبعاد الإسطوانة، العلاقة بين الحافات الخ. هذا المستوى مهم جداً حيث ان الإسطوانات المُعمّمة هي اساس تمثيل الأجسام في النظام. خصائص الإسطوانة المعينة بقيت منفصلة عن الخصائص السطحية الأكثر عمومية.

مستوى السطوح

-- يصف نوع السطح، معلومات حد الحافة، وعلاقة السطح المكانية.

مستوى الحافة

-- يُسجل أنواع الحافت (خطوة، مقعرة محدبة الخ.).

طريقة Lozano-Perez و Grimson (1984)

هذه الطريقة صُممت أصلاً لم طابقة النقاط السطحية الموجهة المتناثرة. لكن أيضاً مُدّدت لم طابقة الحافات المستقيمة من قبل موراي (1988) وآخرون. هنا نناقش الطريقة من ناحية مُطابقة الحافات ذات الخطوط الهباشرة في المشهد مع الحافات ذات الخطوط المباشرة من النموذج، بالاستناد على عمل موراي، وحيث ان الحافات مناسبة أكثر من النقاط عند توفر بيانات العمق الكثيفة. على أية حال، المناقشة صحيحة على حد سواء لحالة النقاط السطحية الموجهة. إن الفرق الوحيد بأن النقاط نطابق إلى الوجوه النموذجية المستوية وتطبيق بعض القيود الهندسية مختلفة قليلاً. تتطلب الطريقة الفرضيات التالية:

- بيانات الصورة قُسمت لتزويد الحافات الدقيقة مع نضمن بعض الخطأ المحدد. هذا الخطأ محدد من ناحية وضع النقاط الأخيرة و الحيرة بالنسبة لإتجاه الحافة. أخطاء النقطة السطحية الموجهة محددة من ناحية حجم صغير من الخطأ.
- الأجسام التي سنعترف يُمكن أن تُشكل ك polyhedra.

هذه الطريقة منقسمة إلى مرحلتين:

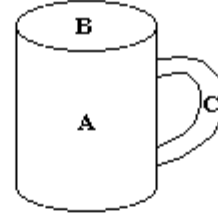
1. تُنتج مجموعة شاملة من التفسيرات العملية للبيانات المُحسنة. التفسيرات تشمل أزواج كل حافة مُحسنة مع بعض حافت النموذج المخزنة. التفسيرات المتناقضة مع قيود الحافة المحلية والتي إشتقت من النموذج تُبدأ ببساطة. قيود الحافة محلية حيث انهم مُستقون من معرفة الحافات المُجاورة فقط وليسوا من المعرفة العامة المُستنبت من وجهة النظر الكاملة أو نموذج الجسم.
2. كل تفسير عملي مُجرب للإتساق بنموذج الجسم. التفسير جائز إذا هو يسمح بالهوران والتحريك الذي يرضع كل حافة مُحسنة في المطابقة مع حافة النموذج مع بعض الخطأ المحدد.

بحث المخطط

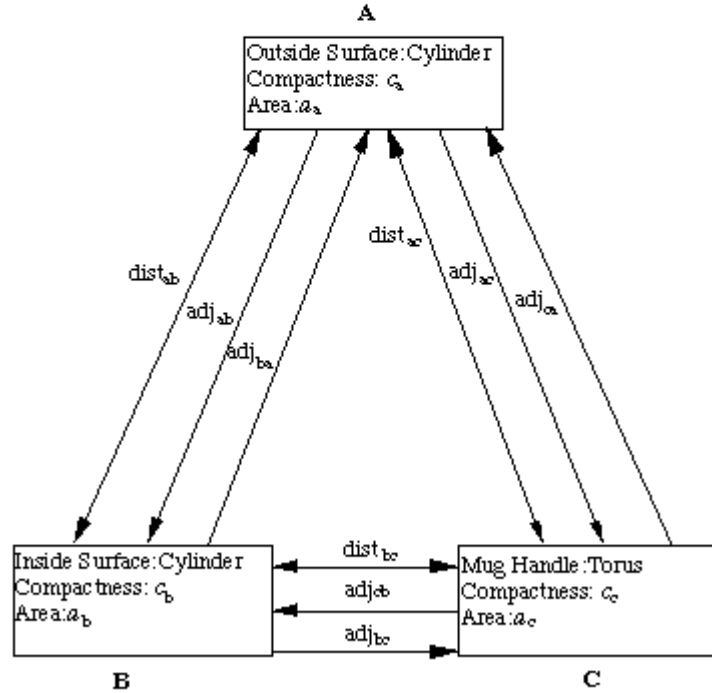
- نموذج الجسم وميزّاتِ المشهدِ مُمَثَّلَة في تركيب مبني على اساس شبكة سيمانطيقية. في الطريقةِ الأصليةِ من قبل Nevatia و Binford دُعوا الشبكة - تركيب المخطط العلائقي:
 \circ هذه الطريقة كانت أساس العديد من الطرق المشهورة للتُمثِيل و تمييز الأجسام في رؤية الحاسوب.

المخطط يَشرِّقُ على :

- مجموعة العُقَدِ متصلة برباط (أيضاً مسمّاة الحافات أو الأقواس).
- تُمَثَّلُ كُلُّ عَقْدَة ميزّة جسم (على سبيل المثال ، سطح)
- العُقَدُ يُمكنُ أَنْ تُعرَّفَ بعدّة مِنْ خصائص الميزّات (مثل الحجم، الشكل، المنطقة ، نوع السطح الخ.).
- تُمَثَّلُ صلات المخطط العلائقَاتِ بين الميزّاتِ - ومثال على ذلك: -
 \circ مسافة بين المراكز للميزّات،
 \circ جوار الميزّات -- نسبة طول الحدّ المشترك بين الميزّتين إلى طول حافةِ أولاً مسمّى الميزّة.
- نموذج تمثيل حدّ يُمكنُ أَنْ يُمَثَّلَ كهذا النوع مِنْ مخطط الجسم.



(a) Picture of a mug



(b) Relational Graph of Mug

الرسم 49. صورة فنجان وتمثيل مخططها البسيط

نلاحظ ان بعض العلاقات هي مزدوجة الاتجاه، مثل المسافة، في تلك العلاقة لا نَعتمدُ على إتجاهِ الوصلة. العلاقات الأخرى، مثل الجوار، يعتمدُ على الإتجاه.

التعرف:

- مسألة مُجَاراة رسمين بيانيين -- رسم بياني من نموذج الجسم إلى الرسم البياني من المشهد الذي يحتوي الجسم.
- طرق مماثلة يجب أن تأخذ في الحسبان الامتصاص وتداخل الأجسام.
- الوسم بياني المشتق من نموذج صلب من الفجاء يحتوي القاع الذي يتغيب في المشهد (ونموذج المشهد).
- لذا المشكلة في إيجاد رسم بياني ثانوي من الرسم البياني الكامل المشتق من النموذج الصلب.
- هذه مشكلة كبيرة لبحث فضاء - استعمال القيود.

دعنا نعود لمعرفة كيف يؤدي Binford و Nevatia إستراتيجية الم تطابق. يستعملون نموذج سطح بالإسطوانات المعممة كميزات، ترتب في مخطط علائقي. إن الميزات المجمعة في المجموعات تسمى الأشرطة التي تحتوي على مجموعات الحافات الثلاثية الأبعاد. كل مجموعة تُفرض لتشكل حد مفرد لاسطوانة م عممة. المنطقة حيث أن عدة أشرطة مرتبطة مثلت بوصلة. تمثيل كل وصلة يحتوي على قائمة الأشرطة المرتبطة. هذه القائمة منظمّة، بحيث نصنع الأشرطة الأطول أو الأوسع أولاً. لكل منظر جسم هناك مخطط علائقي مبني بالوصلات كعقد، وبالوصلات يُقابل أشرطة مرتبطة. تُخزن هذه الصلات خصائص هندسية مختلفة من الأشرطة، من ضمن ذلك طول المحور، عرض مقطع عرضي متوسط وإطالة وصنف (المخروط أو الإسطوانة).

المخطط الذي يُمثل مشهد يطابق مع وصف النموذج في خطوتين:

1. لكل شريط بارز ثلاث خصائص تختار قيادة المُجَاراة. هذه:

o ربط الشريط،

o نوع الشريط -- طويل أو عريض،

o صنف الشريط -- مخروطي أو إسطواني.

المطابقات الأولية مطلوبة للبحث عن الميزات البارزة في الصورة. إذا عُثر على أكثر من م تطابق واحدة فإن الخصائص أعلاه تُستعمل لتقرر أي المطابقات يجب أن تؤخذ للاعتبار أولاً.

2. إن المطابقات نشأت لضمين الأشرطة الأخرى طالما إتساق علاقات الربط مُحفَظ ب ه. مخطط شكل من وجهة نظر، يسمح لمُجَاراة مخطط النموذج إذا بعض الأشرطة في مخطط النموذج ليست موجود في وجهة النظر.

الخاتمة

قارني العزيز، انت الان تنتهي من قراءة كتاب عن أنظمة الذكاء الاصطناعي. أنا أعني انه خارج إطار هذا الكتاب مازالت هناك قارات من عالم الذكاء الاصطناعي تنتظر مكتشفها من: باحثين، محللين، مطورين و طبيعياً قراء. بسبب ندرة المؤلفات باللغة العربية في هذا المجال، وقلة التخصص به في بلادنا، دفعني للعمل على تأليف هذا الكتاب. مادة هذا الكتاب توضح بأنه، في الوقت الراهن، في مجال أنظمة الذكاء الاصطناعي تتوفر قاعدة نظرية ضخمة، كما تتوفر مجموعة عريضة من مبادئ وتقنيات تطوير هذا النوع من الانظمة. الكثير من هذه المبادئ والتقنيات مدعومة بإدوات البرمجة والتطوير. والهدف الرئيسي للمؤلف من وراء هذا الكتاب، هو تقديم هذه المعلومات للقارئ العربي بلغته و بطريقة مبسطة، لسهولة الفهم، وتعريفه بهذا المجال من المعرفة. من الواضح انه لا يمكن إعطاء تفصيل دقيق لكل مجالات الذكاء الاصطناعي بجميع مراحل تطوير انظمتها والطرق المختلفة لذلك بكتاب واحد. لذلك نجد ان بعض اقسام هذا الكتاب لم تأخذ حقها من التفصيل على اعتبار بأن المؤلف يضع بخطته، انشاء الله، تأليف كتب مختصة بهذه المجالات. أوردت عدد كبير من المراجع التي يمكن للقارئ العودة لها ومعرفة معلومات مفصلة أكثر للجزئية التي تهتمه.

من الواضح للقارئ بعد قراءة هذه الكتاب، ان تطوير أنظمة الذكاء الاصطناعي، عملية شاقة وطويلة (تحتاج احيانا لإعوام) ومكلفة (تقدر بمئات الالاف من الدولارات وحتى الملايين) وتحتاج لفريق عمل بحثي، يعمل اعضاءه مع بعضهم البعض ويملكون الخبرات العالية كل في مجاله. هذا يفسر غلاء هذه الانظمة، وتخصصها بمجال معين، وعدم توفر أنظمة لحل طيف واسع من المشاكل وللاستخدام من قبل شريحة عريضة من المستخدمين. عليه تبقى مسألة إكتشاف النظريات والمبادئ والتقنيات الخاصة بتطوير أنظمة الذكاء الاصطناعي، بمختلف مهام مراحلها كالبحث، التمثيل، التفكير والاستنتاج والرؤية... الخ مسألة حيوية للبحث العلمي بهدف التطوير في مجال الذكاء الاصطناعي. كذلك تبقى عملية تطوير أنظمة الذكاء الاصطناعي على الاسس والمبادئ والنظريات المتوفرة عملية ذات فائدة للانسان بتنفيذ مختلف المهام الحياتية. المسألة الاكثر أهمية، من وجهة نظري، هي نشر هذه المعلومات النظرية والأنظمة المطورة بقواعد معلوماتها لتكون متاحة للدارس والمستخدم على حد سواء وبلغات مختلفة. في الختام أتمنى، ان تستغل مساهمتي المتواضعة، من قبل الدارسين العرب والمتخصصين او الراغبين في التخصص في هذا المجال، للمتابعة في المضي في هذا الطريق، والوصول الى نجاحات اكبر ليصبح هذا المجال صناعة في بلادنا.

مع تحيات المؤلف والى اللقاء في كتاب قادم

د. بشير علي عرنوس

المراجع

- [Александр Шамис](#), Поведение, восприятие, мышление: проблемы создания искусственного интеллекта, [Едиториал УРСС](#), 2005 г.
- [Стюарт Рассел](#), [Питер Норвиг](#), Искусственный интеллект. Современный подход, Издательство: [Вильямс](#), 2006 г.
- [К. Птицин](#), Алгоритмы искусственного интеллекта на языке PROLOG, Издательство: [Вильямс](#), 2004 г.
- [Аркадий Частиков](#), [Татьяна Гаврилова](#), Разработка экспертных систем. Среда CLIPS, Издательство: [БХВ-Петербург](#), 2003 г.
- А. П. Частиков, Д. Л. Белов, Т. А. Гаврилова, [Разработка экспертных систем. Среда CLIPS](#), Твердый переплет (2003)
- А. И. Башмаков, И. А. Башмаков, [Интеллектуальные информационные технологии](#), Твердый переплет (2005)
- Л. Г. Комарцова, А. В. Максимов, [Нейрокомпьютеры](#), Твердый переплет (2004)
- Г. Б. Евгеньев, [Системология инженерных знаний](#), Твердый переплет (2001)
- Ю. А. Григорьев, Г. И. Ревунков, [Банки данных](#), Твердый переплет (2002)
- В. В. Девятков, [Системы искусственного интеллекта](#), Твердый переплет (2001)
- [Вячеслав Афонин](#), [Владимир Макушкин](#), Интеллектуальные робототехнические системы, Издательство: [Интернет-университет информационных технологий](#), 2005 г.
- В. Л. Афонин, В. А. Макушкин, [Интеллектуальные робототехнические системы](#), Твердый переплет (2005)
- [К. Протасова](#), Искусственный интеллект. Стратегии и методы решения сложных проблем, Издательство: [Вильямс](#), 2003 г.
- А. Л. Шамис, [Поведение, восприятие, мышление: проблемы создания искусственного интеллекта](#), Издательство: [Едиториал УРСС](#), 2005 г.
- А. И. Башмаков, И. А. Башмаков, [Разработка компьютерных учебников и обучающих систем](#), Издательство: [Едиториал УРСС](#), 2003 г.
- Н. Ясницкий, [Введение в искусственный интеллект](#), Издательство: [ДМК пресс](#), 2004 г.
- Е. В. Масленников, [Экспертное знание. Интеграционный подход и его приложение в социологическом исследовании](#), Издательство: [Наука](#), 2001 г.
- Бондарев В.Н., Аде Ф.Г., [Искусственный интеллект: Учебное пособие для вузов](#) Твердый переплет (2002)
- В. К. Финн, [Интеллектуальные системы и общество](#), Издательство: [Синтег](#), 2002 г.
- Ю. Ф. Тельнов, [Интеллектуальные информационные системы в экономике., Учебное пособие](#), Мягкая обложка (2000)
- В. В. Корнеев, А. Ф. Гареев, С. В. Васютин, В. В. Райх, [Базы данных. Интеллектуальная обработка информации](#), Мягкая обложка (2000)
- К.А. Пупков, В.Г. Коньков, [Интеллектуальные системы](#), Мягкая обложка (2003)
- В. А. Филиппов, [Интеллектуальный анализ данных: методы и средства](#), Издательство: [Питер](#), 2000 г.

- Б. Я. Лихтциндер, М. А. Кузякин, А. В. Росляков, С. М. Фомичев, [Интеллектуальные сети связи](#), Издательство: [Питер](#), 2000 г.
- Г. А. Кушнир, [Системы искусственного интеллекта. Лекция](#), Издательство: [Маркетинг](#), 2001 г.
- Вагин В.Н., Головина Е.Ю., Загорянская А.А. и др. [Достоверный и правдоподобный вывод в интеллектуальных системах](#), Книгопечатная продукция (2004)
- А. В. Алексеев, А. Н. Борисов, Э. Р. Вилюмс, Н. Н. Слядзь, С. А. Фомин, [Интеллектуальные системы принятия проектных решений](#), Издательство: [Zinatne](#), 1997 г.
- Продеус А. Н., Захрабова Е. Н. [Экспертные системы в медицине](#), Издательство: [Век +](#), 1998 г.
- Т. А. Гаврилова, В. Ф. Хорошевский, [Базы знаний интеллектуальных систем](#)
Букинистическое издание (2000)
- Т. А. Гаврилова, Искусственный интеллект. В трех книгах. Книга 3 . Программы и аппаратные средства, Издательство: [Радио и связь](#), 1990 г.
- [Вагин В.Н., Головина Е.Ю., Загорянская А.А. и др.](#), Достоверный и правдоподобный вывод в интеллектуальных системах, 2004 г.
- А. В. Алексеев, А. Н. Борисов, Э. Р. Вилюмс, Н. Н. Слядзь, С. А. Фомин, [Интеллектуальные системы принятия проектных решений](#), Твердый переплет (1997)
- Еремеев А.П. Экспертные модели и методы принятия решений. - М.: МЭИ, 1995.
- Загорулько Ю.А., Попов И.Г. Представление знаний в интегрированной технологической среде SemP-TAO// Сб. Проблемы представления и обработки не полностью определенных знаний/ - М. - Новосибирск, RRIAI, 1996.
- Заде Л. Понятие лингвистической переменной/ Пер. с англ. М.: Мир, 1976. - 165 С.
- Искусственный интеллект: - В 3 кн.: Справочники/ Под ред. Э.В. Попова, Д.А. Поспелова, В.Н. Захарова, В.Ф. Хорошевского. - М.: Радио и связь, 1990.
- Першиков В.И., Савинков В.М. Толковый словарь по информатике. - М., Финансы и статистика, 1991.
- Попов Э.В., Фридман Г.Р. Алгоритмические основы интеллектуальных роботов и искусственного интеллекта. - М.: Наука, 1976.
- Попов Э.В. Экспертные системы: Решение неформализованных задач в диалоге с ЭВМ. - М.: Наука. Гл. ред. ф-м лит., 1987.
- Попов Э.В. и др. Реинжиниринг бизнеса: Реинжиниринг организаций и информационные технологии. - М.: Финансы и статистика, 1997.
- Поспелов Г.С. Системный анализ и искусственный интеллект. - М.: ВЦ АН СССР, 1980.
- Поспелов Г.С. Искусственный интеллект - основа новой информационной технологии. - М.: Наука, 1988.
- Поспелов Д.А. Ситуационное управление: теория и практика. - М.: Наука, 1980.
- Минский М. Структура для представления знания // Психология машинного зрения/ - М.: Мир, 1978.
- Уотерман Д. Руководство по экспертным системам. - М.: Мир, 1989.
- Швецов А.Е. Основные положения технологии активных объектов. - Новосибирск, Препринт/ Рос. НИИ ИИ, 1995.

- Александров Н.И. и др. Моделирование организации и управления решением научно-технических проблем. М., Наука, 1988.
- Абчук В.А., Бункин В.А. Интенсификация: принятие решений. - Л.: Лениздат, 1987. - 174 С.
- Борисов А.Н. и др. Модели принятия решений на основе лингвистической переменной. - Рига; ЗИНАТНЕ, 1982. С. 173
- Ефимов Е.И. Решатели интеллектуальных задач. - М.: Наука, 1982.
- Дерягин Р.И. Алгоритм решения исследовательских проблем //Информатика и ее проблемы. - Новосибирск, Наука, вып.5, 1972.
- Жуковин В.Е. Многокритериальные модели принятия решений с неопределенностью. - Тбилиси, 1983.
- Ларичев О.И. Наука и искусство принятия решений. М., 1979.
- Ларичев О.И. Проблемы принятия решений с учетом факторов риска и безопасности. Вестник АН СССР, 1987, N 11, С. 38-45.
- Лотов А.В. и др. Компьютер и поиск компромисса. Метод достижимых целей. М. Наука, 1997.
- Макаров И.М. и др. Теория выбора и принятия решений. М.: Наука, 1987.
- Ниссинен Й., Воутилайнен Э. Время руководителя: эффективность использования.- М.: Экономика. 1988. -193 С.
- Озерной В.М. Принятие решений (обзор). - "Автоматика и телемеханика", N11, 1971, С.106-121.
- Основы функционально - стоимостного анализа. / Под ред. М.Г. Карпунина и Б.И. Майданчика. - М.: Энергия, 1980.-176 С.
- Орловский С.А. Проблемы принятия решений при расплывчатой информации. - М.: наука, 1981.
- Саати т. Принятие решений: Метод анализа иерархий/ Пер. с англ. - М.: Радио и связь, 1993.
- Функционально-стоимостный анализ в электротехнике/ Под ред. М.Г. Карпунина. М.: Энергоатомиздат, 1984. - 288 С.
- Френкс Л. Теория сигналов. - М.: Сов. радио, 1974.
- Ennals, R., *Artificial Intelligence: Applications to Logical Reasoning and Historical Research*. New York: Wiley, 1985.
- Banerji, R. B. (1980). *Artificial Intelligence: A Theoretical Approach*. North Holland, NewYork.
- Winston, P. H., *Artificial Intelligence*. second edition. Reading, Mass.: Addison-Wesley, 1984.
- Charniak, E. and McDermott, D., *Introduction to Artificial Intelligence*. Reading, Mass.: Addison-Wesley, 1985.
- Gevarter, W. B., *Intelligent Machines: An Introductory Perspective of Artificial Intelligence and Robotics*. Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1985.
- Reitman, W., (ed.), *Artificial Intelligence Applications for Business*. Norwood, N. J.: Ablex, 1984.
- Feigenbaum, E., and McCorduck, P., *The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World*. Reading, Mass.: Addison-Wesley, 1983.
- Waterman, D., *A Guide to Expert Systems*. Reading, Mass.: Addison-Wesley, 1986. Hayes-Roth, F., Lenat, D., and Waterman, D., (eds.), *Building Expert Systems*. Reading, Mass.: Addison-Wesley, 1983.
- Winograd, T., *Language As a Cognitive Process, Volume I: Syntax*. Reading, Mass.: Addison-Wesley, 1983.
- Schank, R. and Abelson, R., *Scripts, Plans, Goals, and Understanding*. Hillsdale, N.J.: Lawrence Erlbaum, 1977.
- Ballard, D. and Brown, C., *Computer Vision*. Englewood Cliffs, N.J.: Prentice-Hall, 1982.

- Wos, L., Overbeek, R., Lusk, E., and Boyle, J., *Automated Reasoning: Introduction and Applications*. Englewood Cliffs, N. J., Prentice-Hall, 1984.
- Maier, D., "Databases in the Fifth Generation Project: Is Prolog a Database Language?", in *New Directions for Database Systems*, Ariav, G. and Clifford, J., eds. Norwood, N.J.: Ablex, 1986,
- Fishman, D., "The DBMS--Expert-System Connection", in *New Directions for Database Systems*, Ariav, G. and Clifford, J., eds. Norwood, N.J.: Ablex, 1986,
- Michalski, R., Carbonell, J., and Mitchell, T., eds., *Machine Learning: An Artificial Intelligence Approach*. Palo Alto, Ca.: Tioga, 1983.
- Goldberg, A., and Robson, D., *Smalltalk-80: The Language and Its Implementation*. Reading, Mass.: Addison-Wesley, 1983.
- Winston, P., and Horn, B., *Lisp*, second edition. Reading, Mass.: Addison-Wesley, 1984.
- Olivier Faugeras (1993). *Three-Dimensional Computer Vision, A Geometric Viewpoint*, MIT Press.
- David A. Forsyth and Jean Ponce (2003). *Computer Vision, A Modern Approach*, Prentice Hall
- Richard Hartely and Andrew Zisserman (2003). *Multiple View Geometry in computer vision*, Cambridge University Press
- Aha, D. W., Kibler, D., and Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, 6(1):37–66.
- Al-Attar, A. (1991). *Structured Decision Tasks Methodology*. Attar Software Ltd., Newlands
- Aleksander, I., Thomas, W. V., and Bowden, P. A. (1984). Wisard: A radical step forward in image recognition. *Sensor Review*, 4:120–124.
- Arbab, B. and Michie, D. (1988). Generating expert rules from examples in prolog. In
- Hayes, J. E., Michie, D., and Richards, J., editors, *Machine Intelligence 11*, pages 289 –304. Oxford University Press, Oxford.
- Bain, M. (1990). Machine-learned rule-based control. In Grimbale, M., McGhee, J., and Mowforth, P., editors, *Knowledge-Based Systems in Industrial Control*, pages 222–244, Stevenage. Peter Peregrinus.
- Bilbro, G. and den Bout, D. V. (1992). Maximum entropy and learning theory. *Neural Computation*, 4:839–853.
- Bratko, I. (1991). Qualitative modelling: Learning and control. In *Proceedings of the 6th Czechoslovak Conference on Artificial Intelligence*. Prague.
- Bratko, I. Mozetic, I. and Lavrac, L. (1989). *KARDIO: A Study in deep and Qualitative Knowledge for Expert Systems*. MIT Press, Cambridge, MA, and London.
- Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, Ca.
- Buntine, W. (1992). Learning classification trees. *Statistics and Computing*, 2:63–73.
- Dvorak, D. L. (1987). Expert systems for monitoring and control. Technical Report
- Ersoy, O. K. and Hong, D. (1991). Parallel, self-organizing, hierarchical neural networks for vision and systems control. In Kaynak, O., editor, *Intelligent motion control: proceedings*
- Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*. Academic Press, 2nd edition.
- Geman, S. (1992). Neural networks and the bias/variance dilemma. *Neural computation*, 4:1–58.
- Hecht-Nielsen, R. (1989). *Neurocomputing*. Addison-Wesley, Reading, Mass.
- Hertz, J., Krogh, A., and Palmer, R. (1991). *Introduction to the Theory of Neural Computation*. Addison-Wesley.
- Kohonen, T., Barna, G., and Chrisley, R. (1988). Statistical pattern recognition with neural networks: Benchmarking studies. In *IEEE International Conference on Neural Networks*, volume 1, pages 61–68, New York. (San Diego 1988), IEEE.
- Krishnaiah, P. and Kanal, L., editors (1982). *Classification, Pattern Recognition, and Reduction of Dimensionality*, volume 2 of *Handbook of Statistics*. North Holland, Amsterdam.

- Clocksin, W. and Mellish, C., *Programming in Prolog*. second edition. Berlin, B.R.D.: Springer-Verlag, 1984. A well-written introduction to Prolog as a programming language, including features not covered in our book. Uses a dialect containing ours. Sterling, L. and Shapiro, E., *The Art of Prolog*. Cambridge, Mass.: MIT Press, 1986. A more detailed introduction to Prolog.
- Bratko, I., *Prolog Programming for Artificial Intelligence*. Reading, Mass.: Addison-Wesley, 1986.
- Clark, K., and McCabe, F., *Micro-Prolog: Programming in Logic*. Englewood Cliffs, N. J.: Prentice-Hall International, 1984.
- Dudgeon, D.E. and R.M. Mersereau, *Multidimensional Digital Signal Processing*. 1984, Englewood Cliffs, New Jersey: Prentice-Hall.
- Castleman, K.R., *Digital Image Processing*. Second ed. 1996, Englewood Cliffs, New Jersey: Prentice-Hall.
- Oppenheim, A.V., A.S. Willsky, and I.T. Young, *Systems and Signals*. 1983, Englewood Cliffs, New Jersey: Prentice-Hall.
- Papoulis, A., *Systems and Transforms with Applications in Optics*. 1968, New York: McGraw-Hill.
- Russ, J.C., *The Image Processing Handbook*. Second ed. 1995, Boca Raton, Florida: CRC Press.
- Giardina, C.R. and E.R. Dougherty, *Morphological Methods in Image and Signal Processing*. 1988, Englewood Cliffs, New Jersey: Prentice-Hall. 321.
- Gonzalez, R.C. and R.E. Woods, *Digital Image Processing*. 1992, Reading, Massachusetts: Addison-Wesley. 716.
- Goodman, J.W., *Introduction to Fourier Optics*. McGraw-Hill Physical and Quantum Electronics Series. 1968, New York: McGraw-Hill. 287.
- Stockham, T.G., *Image Processing in the Context of a Visual Model*. Proc. IEEE, 1972. **60**: p. 828 - 842.
- Murch, G.M., *Visual and Auditory Perception*. 1973, New York: Bobbs-Merrill Company, Inc. 403.
- Frisby, J.P., *Seeing: Illusion, Brain and Mind*. 1980, Oxford, England: Oxford University Press. 160.
- Born, M. and E. Wolf, *Principles of Optics*. Sixth ed. 1980, Oxford: Pergamon Press.
- Young, I.T., *Quantitative Microscopy*. IEEE Engineering in Medicine and Biology, 1996. **15**(1): p. 59-66.
- Dorst, L. and A.W.M. Smeulders, *Length estimators compared*, in *Pattern Recognition in Practice II*, E.S. Gelsema and L.N. Kanal, Editors. 1986, Elsevier Science: Amsterdam. p. 73-80.
- Vossepoel, A.M. and A.W.M. Smeulders, *Vector code probabilities and metrication error in the representation of straight lines of finite length*. Computer Graphics and Image Processing, 1982. **20**: p. 347-364.
- Groen, F.C.A., R.J. Ekkers, and R. De Vries, *Image processing with personal computers*. Signal Processing, 1988. **15**: p. 279-291.
- Verbeek, P.W., H.A. Vrooman, and L.J. Van Vliet, *Low-Level Image Processing by Max-Min Filters*. Signal Processing, 1988. **15**: p. 249-258.
- Young, I.T. and L.J. Van Vliet, *Recursive Implementation of the Gaussian Filter*. Signal Processing, 1995. **44**(2): p. 139-151.
- Vincent, L., *Morphological transformations of binary images with arbitrary structuring elements*. Signal Processing, 1991. **22**(1): p. 3-23.
- Van Vliet, L.J. and B.J.H. Verwer, *A Contour Processing Method for Fast Binary Neighbourhood Operations*. Pattern Recognition Letters, 1988. **7**(1): p. 27-36.
- Canny, J., *A Computational Approach to Edge Detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986. **PAMI-8**(6): p. 679-698.
- Marr, D. and E.C. Hildreth, *Theory of edge detection*. Proc. R. Soc. London Ser. B., 1980. **207**: p. 187-217.
- Verbeek, P.W. and L.J. Van Vliet, *On the Location Error of Curved Edges in Low-Pass Filtered 2D and 3D Images*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1994. **16**(7): p. 726-733.

- Lee, J.S.L., R.M. Haralick, and L.S. Shapiro. *Morphologic Edge Detection*. In *8th International Conference on Pattern Recognition*. 1986. Paris: IEEE Computer Society.
- Van Vliet, L.J., I.T. Young, and A.L.D. Beckers, *A Non-linear Laplace Operator as Edge Detector in Noisy Images*. *Computer Vision, Graphics, and Image Processing*, 1989. **45**: p. 167-195.
- Meyer, F. and S. Beucher, *Morphological Segmentation*. *J. Visual Comm. Image Rep.*, 1990. **1**(1): p. 21-46.

هذا الكتاب

الذكاء الاصطناعي جزء من علم الحاسبات والذي يهتم بأنظمة الحاسوب الذكية، تلك، الأنظمة التي تمتلك الخصائص المرتبطة بالذكاء واتخاذ القرار والمثابرة لدرجة ما لسلوك البشري في هذا المجال – فيما يخص اللغات، تمثيل المعلومات، البحث، التعلم، التفكير، اتخاذ القرار وحل المشاكل... الخ.

هذا الكتاب موجه للدارسين والباحثين باللغة العربية والعاملين والمطورين والمبرمجين في مجالات تقنية المعلومات من هندسة الكمبيوتر ونظم المعلومات. إن أسلوب الملخصات المعتمد في هذا الكتاب، ليكون أقرب الى الكتاب الأكاديمي، ويكون اسهل للقراءة والفهم من أسلوب السرد يسمح للدارسين في المجالات الاخرى كالهندسة، الطب، المجالات العلمية الأخرى كالرياضيات، الفيزياء، الكيمياء، الجغرافيا، علوم الارض، علوم الفضاء... الخ، كذلك العلوم الاجتماعية بأنواعها بقراءة وفهم المضمون العام لهذا الكتاب.